RI 8479

Bureau of Mines Report of Investigations/1980

NEWARK PUBLIC LIBRARY

JAN 8 1981

U. S. GOVT. PUBS.
DEPOSITORY 376

Computer Modeling of Fluid Flow During Production and Environmental Restoration Phases of In Situ Uranium Leaching

By Robert D. Schmidt

UNITED STATES DEPARTMENT OF THE INTERIOR

I 28.23:

Report of Investigations 8479

Computer Modeling of Fluid Flow During Production and Environmental Restoration Phases of In Situ Uranium Leaching

By Robert D. Schmidt

UNITED STATES DEPARTMENT OF THE INTERIOR
Cecil D. Andrus, Secretary

BUREAU OF MINES
Lindsay D. Norman, Director

READEX FICHE

This publication has been cataloged as follows:

Schmidt, Robert D

Computer modeling of fluid flow during production and environmental restoration phases of in situ uranium leaching.

(Report of investigations - Bureau of Mines ; 8479)
Bibliography: p. 29.
Supt. of Docs. no.: I 28.23:8479.

1. Uranium mines and mining–Mathematical models. 2. Uranium mines and mining–Data processing. 3. Solution mining–Mathematical models. 4. Solution mining–Data processing. 5. Fluid dynamics–Mathematical models. 6. Fluid dynamics–Data processing. I. Title. II. Series: United States. Bureau of Mines. Report of investigations ; 8479.

TN23.U43 [TN490.U7] 622s [622'.34932] 80-607904

CONTENTS

	Page
Abstract	1
Introduction	1
General description	1
Hydrology simulation	2
Well-aquifer geometry and nomenclature	2
Computational methods	4
General model	4
Directional permeability	6
Fluid velocity	8
Streamline development	8
Partial screening	10
Areal sweep	11
Aquifer assumptions	12
Model verification and validation	12
Scientific verification	12
Model assumptions and general validation	13
Site-specific model revalidation and calibration	14
User options and input requirements	14
Input options	14
Output options	15
Principal input parameters	15
Program structure	15
Computer program accessibility	18
Applications of ISL-50	18
Case 1.--Three stages of site operation	18
Case 2.--Screen versus wellhead locations	20
Case 3.--Directional anisotropy	23
Case 4.--Flow velocity contours	23
Case 5.--Three-dimensional simulation	23
Case 6.--Five-spot confinement	27
Conclusions	28
References	29
Appendix A.--Input variable description	30
Appendix B.--Sample input data	33
Appendix C.--Sample output	34
Appendix D.--ISL-50 program listing	35

ILLUSTRATIONS

1.	Well-aquifer cross section	4
2.	Single-well incremental stream tube area, A and B	11
3.	Program flow of control	17
4.	Streamline pattern, site development stage	19
5.	Streamline pattern, operational stage	19
6.	Streamline pattern, fluid excursions	21
7.	Streamline pattern, restoration stage	21
8.	Five-spot pattern using screen locations	22
9.	Five-spot pattern with directional permeability	22

ILLUSTRATIONS--Continued

Page

10. Streamline pattern and isovelocity contour..... 23
11. Partially screened aquifer, 20-ft injection screen..... 24
12. Partially screened aquifer, 10-ft injection screen..... 25
13. Plan view, 20-ft injection screen..... 26
14. Plan view, 10-ft injection screen..... 26
15. Isolated five-spot and five-spot with four guard wells..... 27
16. Streamline pattern, aquifer-oriented screens..... 28

TABLE

1. Program components..... 16

COMPUTER MODELING OF FLUID FLOW DURING PRODUCTION AND ENVIRONMENTAL RESTORATION PHASES OF IN SITU URANIUM LEACHING

by

Robert D. Schmidt [1]

ABSTRACT

This Bureau of Mines report describes the development and application of a computer model for simulating the hydrological activity associated with in situ leaching. The model is intended to provide uranium resource developers with a description of the flow behavior of leachants and ground water during the development, production, and restoration phases of a leaching operation involving an arbitrary pattern of injection and recovery wells.

Different aquifer environments are modeled, using a closed-form solution to the partial differential equation that describes three-dimensional changes in piezometric head as a result of pumping from leachant injection and recovery wells. The computer program can model a maximum of 50 arbitarily located wells.

Numerical techniques involving difference quotients and Taylor expansions about time points are used to derive time, velocity, areal sweep, and fluid volume parameters associated with leaching hydraulics. These parameters are output by the program in graphic and tabular formats. Other numeric methods insure that the program running time is minimized without significantly affecting the accuracy of results.

INTRODUCTION

General Description

In recent years, increased demand for uranium has focused interest on the extraction of uranium from low-grade ore deposits. In situ leaching is one extraction method that shows promise of being both economically and environmentally advantageous.

This Bureau of Mines report describes a computer model, intended primarily for uranium resource developers, which simulates the hydrological activity associated with in situ leaching. The simulated fluid flow that results from

[1]Operations research analyst, Twin Cities Research Center, Bureau of Mines, Twin Cities, Minn.

user selection of operating parameters such as well configuration, screen penetration, and pumping rate, and of geohydrologic parameters such as aquifer type, directional permeability, and ground water flow is presented graphically and tabularly.

The model was developed directly from a three-dimensional closed-form solution to the aquifer hydrology equation presented by Hantush and Jacob (8, 10).[2] This solution describes change in fluid pressure as a result of injection and recovery wells that partially screen an aquifer bounded above and below by impermeable layers. Well configurations and pumping rates are user specified, and up to 50 wells may be simulated.

User input requirements have been simplified as much as possible within the confines of necessary model flexibility. Throughout this report, the author's intent has been to develop a predictive tool that can be readily applied to predict fluid movement resulting from site-specific operating parameters.

Hydrology Simulation

Simulation of fluid movement under a variety of operating and site-specific conditions can have application to both the production and environmental concerns of in situ developers. The ISL-50 model enables an assessment of the effectiveness and acceptability of various well and pumping configurations, in terms of both fluid dispersion within the ore pod and fluid excursions into the surrounding aquifer. The hydrology of ground water restoration measures may also be simulated.

In addition, simulating the hydrology associated with in situ leaching is essential for analysis of more complex modeling problems involving chemical concentration and leaching chemistry kinetics.

WELL-AQUIFER GEOMETRY AND NOMENCLATURE

In this model, the recharge and discharge wells have been idealized as line sources or sinks. Wells are constructed to intersect a confined aquifer, which may be leaky or nonleaky (fig. 1). A leaky aquifer is slightly permeable; the permeability is assumed to be totally vertical from an aquitard layer, that is, a semipervious confining bed, either above or below, but usually above it.

The definitions and nomenclature presented apply to the subsequent account of hydrology equation application. As far as possible, these variable names were used in the FORTRAN computer program listed in appendix D. The important variables are as follows:

---

[2]Underlined numbers in parentheses refer to items in the list of refereuces preceding the appendixes.

Δh change in head (pressure) from the original piezometric surface, in feet.

(x_i, y_i) Cartesian coordinates of reference frame--x and y are in the horizontal plane of the aquifer, and z is the vertical distance from the top of the aquifer, in feet (fig. 1).

r_i radius of the point (x,y,z) from the center of well i, in feet (fig. 1) $r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$.

t time interval of evaluation, in hours.

K aquifer permeability, in feet per second.

K' aquitard permeability, in feet per second.

b aquifer thickness, in feet (fig. 1).

b' aquitard thickness, in feet (fig. 1).

B leakage factor $= \sqrt{\frac{b \cdot b' \cdot K}{K'}}$, in feet.

γ specific weight of fluid, in pounds per cubic foot.

α compressibility of solid skeleton, in square feet per pound.

β compressibility of fluid, in square feet per pound.

n porosity, dimensionless.

Q_i input (-) or output (+) flow rate from well i, in cubic feet per second.

r_e radius of influence (radius from each well at which no drawdown occurs, default = 600), in feet.

ℓ number of operating wells in the pattern.

D distance from the top aquifer confining layer to the screen top, in feet (fig. 1).

L distance from the top aquifer confining layer to the screen bottom, in feet (fig. 1).

s $\gamma b(\alpha + n\beta)$ storage coefficient.

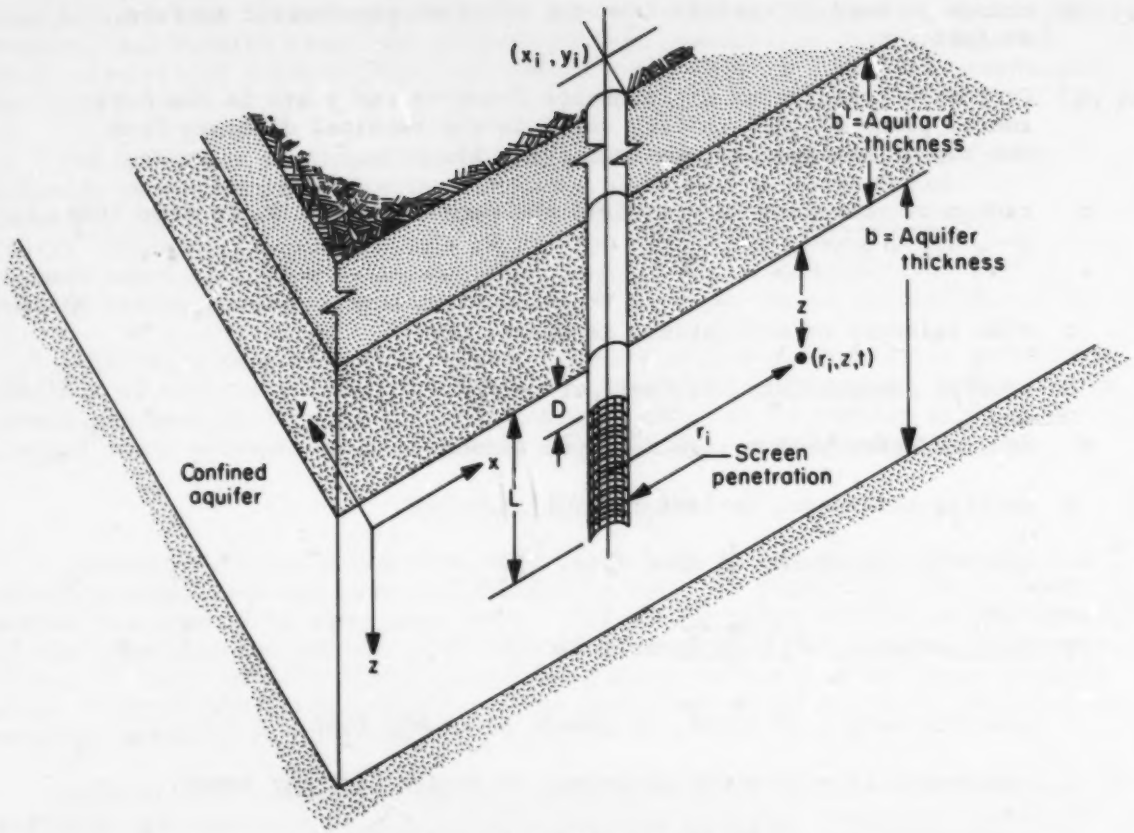

FIGURE 1. - Well-aquifer cross section.

COMPUTATIONAL METHODS

General Model

Hantush (7-8) has developed one of the more general models in the hydrology literature. This model describes three dimensional time-dependent flow resulting from pumped wells in a leaky, confined aquifer, where the aquifer thickness is only partially screened by the wells (fig. 1). The nonleaky confined aquifer, the confined aquifer with completely penetrating well screens, and steady-state versions of the above are all special cases covered by Hantush's more general model.

A statement of the general flow problem, boundary conditions, and the solution derived to model fluid flow was presented in an earlier publication (11). However, the solution was derived exclusively for the isotropic aquifer and symmetric five-spot well pattern. Generalization to the multiple-well problem with arbitrary well locations is straightforward and simply extends solution additivity and image well techniques used in the earlier version from 5 to as many as 50 wells. Generalization to the directionally anisotropic case is detailed below.

For one well, the general flow problem with boundary conditions is given by

$$\frac{\partial^2 \Delta h}{\partial r^2} + \frac{1}{r}\frac{\partial \Delta h}{\partial r} + \frac{1}{r^2}\frac{\partial^2 \Delta h}{\partial \theta^2} + \frac{\partial^2 \Delta h}{\partial z^2} - \frac{\Delta h}{B^2} = \frac{s}{K}\frac{\partial \Delta h}{\partial t} \tag{1}$$

where

$$\Delta h = \Delta h\ (r,\theta,z,t),$$

$$\Delta h\ (r,\theta,z,0) = 0,$$

$$\Delta h\ (\infty,\theta,z,t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\ (r,\theta,0,t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\ (r,\theta,b,t) = 0,$$

and

$$\lim_{r \to 0} 2\pi K r \int_0^b \frac{\partial \Delta h}{\partial r}\, dz = -Q. \tag{2}$$

Equation 1, with boundary conditions 2, describes flow from a well in an isotropic aquifer. In such cases, the flow is radial, so equations 1 and 2 reduce to the expression given by Hantush (7), or Kurth and Schmidt (11), as

$$\frac{\partial^2 \Delta h}{\partial r^2} + \frac{1}{r}\frac{\partial \Delta h}{\partial r} + \frac{\partial^2 \Delta h}{\partial z^2} - \frac{\Delta h}{B^2} = \frac{s}{K}\frac{\partial \Delta h}{\partial t} \tag{3}$$

where

$$\Delta h = \Delta h\ (r,z,t),$$

$$\Delta h\ (r,z,0) = 0,$$

$$\Delta h\ (\infty,z,t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\ (r,0,t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\ (r,b,t) = 0,$$

and

$$\lim_{r \to 0} 2\pi K r \int_0^b \frac{\partial \Delta h}{\partial r}\, dz = -Q. \tag{4}$$

Using superposition to determine the effect of ℓ injection and recovery wells, the solution to Hantush's general problem is given by

$$\Delta h\,(r,z,t) = \sum_{i=1}^{\ell} \frac{Q_i}{2\pi Kb}\left[W_i\left(U_i, \frac{r_i}{B}\right) + \frac{2b}{\pi(L-D)} \sum_{m_i=1}^{\infty} \frac{1}{m_i} \cos\left(\frac{m_i \pi z}{b}\right) \right.$$

$$\left. \cdot \left(\sin\left(\frac{m_i \pi L}{b}\right) - \sin\left(\frac{m_i \pi D}{b}\right)\right) \cdot W_i\left(U_i, \sqrt{\left(\frac{r_i}{B}\right)^2 + \left(\frac{m_i \pi r_i}{b}\right)^2}\right)\right] \qquad (5)$$

where
$$U_i = \frac{\gamma(\alpha + \beta n)}{4K} \cdot \frac{r_i^{\,2}}{t}, \qquad (6)$$

$$W_i\,(U_i, \lambda) = \int_{U_i}^{\infty} \frac{1}{y} \exp\left(y - \frac{\lambda^2}{4y}\right) dy, \qquad (7)$$

and
$$i = 1,2,3....\ell.$$

Pumping rates Q_i in equation 5 are positive for recovery wells and negative for injection wells. Solution additivity in equation 5 is a simple, but important, consequence of the linearity of equation 3. That is, if h_i and h_j individually are solutions to equation 3, then $(h_i + h_j)$ is also a solution.

Directional Permeability

A further refinement from the author's earlier model (11) allows the user to specify anisotropy in the form of idealized maximum and minimum permeability values in two principal horizontal directions (4). The principal permeabilities are denoted by K_x' and K_y', where the principal directions are chosen such that the x'-axis is in the direction of maximum permeability K_x', and the and the y'-axis is in the direction of minimum permeability K_y'.

With K_x' and K_y', thus accounted for, the leaky nonsteady flow problem with partially penetrating well screens (bringing in the z-coordinate and K_z) becomes in rectangular coordinate form

$$K_x' \frac{\partial^2 \Delta h}{\partial x'^2} + K_y' \frac{\partial^2 \Delta h}{\partial y'^2} + K_z \frac{\partial^2 \Delta h}{\partial z^2} - \frac{\Delta h}{B^2} = S \cdot \frac{\partial \Delta h}{\partial t} \qquad (8)$$

with
$$\Delta h = \Delta h\,(x,y,z,t),$$

$$\Delta h\,(x,y,z,0) = 0,$$

$$\Delta h\,(\infty,y,z,t) = 0,$$

$$\Delta h\,(x,\infty,z,t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\,(x,y,0,t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\,(x,y,b,t) = 0,$$

and $$\lim_{r \to 0} 2\pi Kr \int_0^b \frac{\partial \Delta h}{\partial r}\, dz = -Q. \tag{9}$$

The substitutions

$$\hat{y} = \sqrt{\frac{K_x'}{K_y'}} \cdot y' \text{ and } \hat{z} = \sqrt{\frac{K_x'}{K_z}} \cdot z \tag{10}$$

transform equation 8 into

$$\frac{\partial^2 \Delta h}{\partial x'^2} + \frac{\partial^2 \Delta h}{\partial \hat{y}'^2} + \frac{\partial^2 \Delta h}{\partial \hat{z}^2} - \frac{\Delta h}{K_x' B^2} = \frac{s}{K_x'} \frac{\partial \Delta h}{\partial t} \tag{11}$$

with initial and boundary conditions 9 expressed in the (x', $\hat{y}'$, $\hat{z}$, t) system.

Writing equation 11 in cylindrical coordinates this becomes

$$\frac{\partial^2 \Delta h}{\partial \hat{r}'^2} + \frac{1}{\hat{r}'} \frac{\partial \Delta h}{\partial \hat{r}'} + \frac{1}{\hat{r}'^2} \cdot \frac{\partial^2 \Delta h}{\partial \theta^2} + \frac{\partial^2 \Delta h}{\partial \hat{z}^2} - \frac{\Delta h}{K_x' B^2} = \frac{s}{K_x'} \cdot \frac{\Delta \partial h}{\partial t} \tag{12}$$

with initial and boundary conditions

$$\Delta h = \Delta h\ (\hat{r}', \theta, \hat{z}, t),$$

$$\Delta h\ (r', \theta, z, 0) = 0,$$

$$\Delta h\ (\infty, \theta, \hat{z}, t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\ (\hat{r}', \theta, 0, t) = 0,$$

$$\frac{\partial \Delta h}{\partial z}\ (\hat{r}, \theta, b, t) = 0,$$

and $$\lim_{\hat{r}' \to 0} 2\pi K\hat{r}' \int_0^b \frac{\partial \Delta h}{\partial \hat{r}'}\, d\hat{z} = -Q, \tag{13}$$

and with the distance $\hat{r}'$ of a point ($\hat{r}'$, θ, z, t) from the center of a well located at the origin in the transformed system given by

$$\hat{r}' = \sqrt{x'^2 + y'^2}. \tag{14}$$

Because equations 12 and 13 are identical to equations 1 and 2, the physically anisotropic case becomes mathematically isotropic in the transformed coordinate system. Therefore, the solution to equations 8 and 9

is obtained by solving equations 3 and 4 in the $(\hat{r}, \theta, z, t)$ coordinate system, subsequently transferring to the (x', y', z, t) system, using the substitutions

$$\sqrt{\frac{K_{x'}}{K_{y'}}} \cdot y' \text{ for } \hat{y}' \text{ and } \sqrt{\frac{K_{x'}}{K_{z'}}} \cdot z \text{ for } \hat{z}$$

and then rotating from (x', y', z, t) back to (x, y, z, t).

Fluid Velocity

Having computed Δh at arbitrary points in the aquifer, fluid velocities in the x-direction (V_x) and y-direction (V_y) are found from Darcy's law to be

$$V_x = -\frac{K}{n}\frac{\partial \Delta h}{\partial x} + G_x \tag{15}$$

and

$$V_y = -\frac{K}{n}\frac{\partial \Delta h}{\partial y} + G_y, \tag{16}$$

in the (x, y, z, t) coordinate system, with analogous representation in the $(x', \hat{y}', \hat{z}, t)$ system, where G_x and G_y are the x- and y-components of natural ground water velocity. Replacing these partial derivatives by centered difference quotients gives the computational expressions

$$V_x = -\frac{K}{n} \cdot \left(\frac{\Delta h(x_{n+1}, y_n) - \Delta h(x_{n-1}, y_n)}{x_{n+1} - x_{n-1}}\right) + G_x \tag{17}$$

$$V_y = -\frac{K}{n} \cdot \left(\frac{\Delta h(x_n, y_{n+1}) - \Delta h(x_n, y_{n-1})}{y_{n+1} - y_{n-1}}\right) + G_y . \tag{18}$$

The denominators of these expressions are differences chosen small enough so as not to significantly affect the accuracy of the computed value of velocity.

Streamline Development

Successive streamline coordinates are obtained in the general case using a three-term Taylor expansion about the time point t_n. The relation between a streamline segment Δd_{n+1} and a time interval Δt_{n+1}, where $\Delta d_{n+1} = d(t_{n+1}) - d(t_n)$ and $\Delta t_{n+1} = t_{n+1} - t_n$ is given by,

$$d(t_{n+1}) = d(t_n + \Delta t_{n+1}) = d(t_n) + d'(t_n) \cdot (t_{n+1} - t_n)$$

$$+ d''(t) \cdot \frac{(t_{n+1} - t_n)^2}{2} + Rd(t) \tag{19}$$

or

$$\Delta d_{n+1} = d'(t_n) \cdot \Delta t_{n+1} + d''(t_n) \cdot \frac{(\Delta t_{n+1})^2}{2} + Rd(t_n). \tag{20}$$

The x- and y-components of Δd_{n+1} are obtained from equation 20 with d = x and d = y, respectively. The velocity term $d'(t_n)$ is either $V_x(t_n)$ or $V_y(t_n)$ found from Darcy's law, and the acceleration term $d''(t_n)$ is approximated by a velocity difference quotient as

$$d''(t_n) = \frac{d'(t_n) - d'(t_{n-1})}{\Delta t_n}. \tag{21}$$

The time interval between evaluated streamline points, Δt_{n+1}, is variable in this model and is made dependent on the velocity change of streamline particles. For accurate streamline development, the coordinates of higher velocity particles must be calculated at more frequent intervals than those of lower velocity particles, because changes in direction and velocity occur more rapidly. The Δt_{n+1} chosen should not be so large as to affect the accuracy of streamline calculation, nor so small as to result in excess program execution time. To estimate the accuracy of streamline calculation (that is, the appropriateness of the Δt_{n+1} used to determine Δd_{n+1} in equation 20), the Taylor's series remainder term $Rd(t_n)$ is calculated using the third derivative $d'''(t_n)$ as

$$Rd(t_n) = d'''(t_n) \cdot \frac{(\Delta t_{n+1})^3}{6}. \tag{22}$$

A first approximation for Δt_{n+1}, denoted $\Delta t_{n+1}^{(1)}$, is made by setting $\Delta t_{n+1}^{(1)} = \Delta t_n$. To decide if this or a subsequent approximation $\Delta t_{n+1}^{(i)}$, i = 2, 3, 4.... is appropriate, the normalized $Rd(t_n)$, given by

$$\tilde{Rd}^{(i)}(t_n) = \frac{Rd(t_n)}{Rd(t_1)} = \frac{d'''(t_n) \cdot (\Delta t_{n+1})^3}{d'''(t_1) \cdot (\Delta t_2)^3} \tag{23}$$

is evaluated for i = 1, 2, 3,....

The procedure adopted is to compare $\tilde{Rd}^{(i)}(t_n)$ with upper and lower bounds E_U and E_L. If $E_L > \tilde{Rd}^{(i)}(t_n) > E_U$, then $\Delta t_{n+1}^{(i)}$ is incremented or decremented accordingly, and $\tilde{Rd}^{(i+1)}(t_n)$ is recalculated from equation 23, using the new $\Delta t_{n+1}^{(i+1)}$. If $E_L \leq \tilde{Rd}^{(1)}(t_n) \leq E_U$, then $\Delta t_{n+1}^{(i)}$ is appropriate and is taken to be Δt_{n+1}.

The ratios denoted by E_L and E_U were chosen so that program execution time is close to being minimized, and the accuracy of streamline calculations is not significantly affected. Experiments with various sets of upper and lower bounds, using representative field data, indicated that for fluid velocities normally encountered, E_L = 3 and E_U = 6 were acceptable bounds; even though they are conservative in calculating more than the minimal number of streamline points.

Under different operating conditions, for example, low pumping rates or a widely spaced well field, both E_L and E_U could be increased to reduce program execution time even further. Experiments with the present values as a baseline is a simple procedure. (See program source code listing, line numbers STRM 1940 and STRM 2010.)

For those streamlines beginning at injection wells, careful selection of the initial time interval Δt_1 is especially important, because the change in particle velocity is greatest during a very short interval after injection. Errors in streamline direction introduced at this point, because of a poor straight-line segment approximation to the true streamline path, are propagated over the entire streamline.

A first approximation of Δt_1, $\Delta t_1^{(1)} = 0.25$ hour is used to obtain the first two streamline points (x_1, y_1) and (x_2, y_2) as follows:

$$d(t_1) = d(t_o) + d'(t_o) \cdot \Delta t_1 \tag{24}$$

and

$$d(t_2) = d(t_1) + d'(t_1) \cdot \Delta t_1 \tag{25}$$

where, as before, x- and y-components are obtained by setting d = x and d = y, respectively

Subsequent approximations of Δt_1 (i.e., $\Delta t_1^{(i)}$, i = 2, 3, 4...) are obtained by requiring that the change in x- and y-velocity components be small at points near the injection well. That is,

$$\frac{d'(t_1)}{d'(t_o)} \geqq \psi \text{ where } d'(t_1) < d'(t_o) \text{ and}$$

where ψ is a number slightly less than 1 (0.95). If either the x or y velocity ratio is less than ψ, then a new, smaller approximation for Δt_1 is formed according to $\Delta t_1^{(i)} = \frac{1}{2}\Delta t^{(i-1)}$, and new, $d(t_1)$ and $d(t_2)$ are calculated using equations 24 and 25.

Partial Screening

Having obtained an (x, y) coordinate along a streamline path, equation 20 can be reapplied to determine the (z) coordinate for those pumping situations in which partial well screening results in a vertical flow component. Here, V_z is given by

$$V_z = \frac{-K_z}{n} \cdot \frac{\partial \Delta h}{\partial z}. \tag{26}$$

The value Δt_{n+1} previously calculated for Δd_{n+1} is used to determine z-coordinates along each streamline path. Because $\frac{\partial \Delta h}{\partial z}$ is generally much smaller than either $\frac{\partial \Delta h}{\partial x}$ or $\frac{\partial \Delta h}{\partial y}$, Δz_{n+1} is approximated simply by

$$\Delta z_{n+1} = z'(t_n) \cdot \Delta t_{n+1}. \tag{27}$$

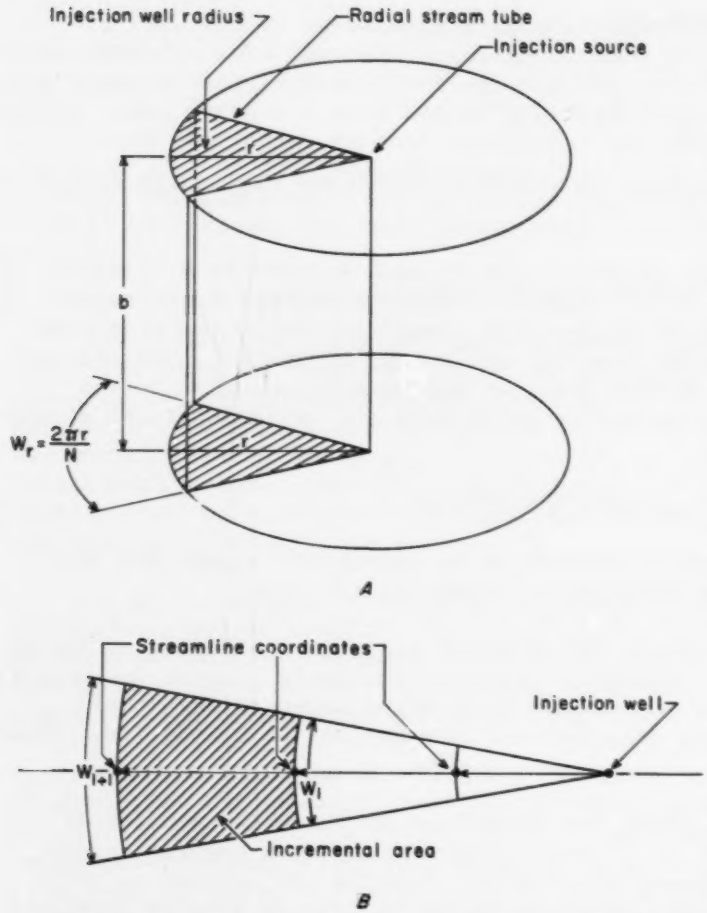

FIGURE 2. - Single-well incremental stream tube area, *A* and *B*.

Areal Sweep

The areal sweep of a well pattern for which fluid flow can be characterized as two-dimensional and steady state is obtained in the ISL-50 program by summing the areal sweeps of individual injection well streamlines.

Because a finite number (N) of radial streamlines are used to approximate flow about an injection well, each streamline can be considered the center of a stream tube, there being then N adjacent stream tubes which together completely fill the flow region (fig. 2A).

Stream tube area is obtained by first noting the continuity condition for steady-state ground water flow. This condition, as applied to successive streamline coordinates, can be expressed as

$$W_j \cdot V_j = W_i \cdot V_i \qquad (28)$$

where W is the stream tube width, and V is the fluid velocity at the ith and jth streamline coordinates.

Taking the injection well radius r as the first streamline coordinate and assuming radial flow at this point, V_r is given by Darcy's law as

$$V_r = \frac{-K}{n} \cdot \left(\frac{\partial \Delta h}{\partial r}\right) = \frac{Q}{2\pi bn} \cdot \left(\frac{1}{r}\right), \qquad (29)$$

and W_r is given as

$$W_r = 2\pi r/N. \qquad (30)$$

Equation 28 can now be rewritten as

$$W_i = \frac{W_r \cdot V_r}{V_i} \tag{31}$$

where V_i is obtained in the general streamline coordinate case using equations 17 and 18.

In the ISL-50 program, W_i is calculated at each successive streamline coordinate and multiplied by the incremental distance between coordinates (fig. 2B). These incremental areas are then summed to obtain the plan view areal sweep of individual streamlines and wells and, ultimately, the entire well pattern. Although this method gives an approximate solution in the three-dimensional case, it is generally valid only for two-dimensional steady-state flow situations.

AQUIFER ASSUMPTIONS

The aquifer being modeled is assumed to be isotropic, except for the limited anisotropy allowed by directional permeability.

Aquifer uniformity is assumed to extend at least to the radius of influence of each well (600 feet is default). However, precisely located impermeable boundaries within this radius can be simulated using standard image well techniques in conjunction with user-defined flow evaluation boundaries. (See Boundary card, appendix A.)

The suitability of this model for situations in which the aquifer is not level and uniformly thick is of some concern since core samples seldom indicate an ore pod of such ideal uniformity. Despite indications of variability, core samples are seldom numerous enough to provide precise knowledge of the variations in upper and lower aquifer boundaries. Therefore, a model should be chosen that is commensurate with the quantity and quality of field data available.

Thus far, all applications of the ISL-50 program have involved aquifers of slightly variable thickness. For these cases, the average thickness was obtained from core samples, and the existence of slight deviations from the horizontal plane was ignored. Although there are obvious limitations, the error introduced by such averaging is probably small compared with other errors resulting from a limited core sampling.

MODEL VERIFICATION AND VALIDATION

Scientific Verification

The flow model used in this program is historically well established. The solution for two-dimensional horizontal flow between a recharge well and a discharge well first appeared in the 1800's (12). Petroleum engineers in the 1930's extended this solution to model two- and three-phase flow of oilfield waterflooding to increase petroleum yield (2).

In the 1940's, hydrologists such as Hantush, began extending earlier hydrodynamics work into the specialized area of three-dimensional flow resulting from multiple wells in various aquifer types and under special conditions. Since then, Hantush's work has been widely accepted and applied in various hydrologic problems. Further scientific verification that Hantush's flow equation does in fact describe ideal fluid flow in a porous medium, owing to recharge and discharge from wells, and given the stated boundary conditions, appears unnecessary for this latest application.

Model Assumptions and General Validation

Apart from verifying Hantush's general flow equation (equation 3) and its numerical solution (equation 5), it is important to determine if the assumptions of aquifer homogeneity incorporated in this model are suitable approximations of the actual conditions at a site. For this model, the principal assumptions of aquifer homogeneity are (1) anisotropic permeability of a directional nature only, (2) a level aquifer of uniform thickness, (3) an aquifer that is everywhere confined, and (4) no large impermeable lenses in the aquifer (clay stringers).

It is impossible to validate the model by examining the flow far underground in a pumped aquifer. At best, only a few engineering measurements can be obtained at selected points in the aquifer, using various pumping tests. Predicted fluid drawdown or predicted front breakthrough time can then be compared with field measurements to establish a degree of model validity.

General validation of the ISL-50 program has thus far involved comparison of predicted and measured breakthrough times at three different pilot leaching sites. In the first two field tests, steady-state front breakthrough time was established by measuring the change in pH at recovery wells following the injection of leachant in the pattern. For the first test, at a pilot five-spot site using a carbonate leach, the field measurement of breakthrough was approximately 3½ days, and the model predicted initial breakthrough after 4 days.

In the second test, attempts to estimate breakthrough time were more difficult. Measurements were made only once each day, which severely limited the experimental accuracy. Also, an undetermined amount of the acid leachant reacted with bicarbonate naturally present in the formation, and this delayed the change in pH at the output well for an unknown period. Field data in this case provided only an upper bound for actual breakthrough time: the program predicted 1½ days for breakthrough, and recovery well pH began to change after approximately 3 days.

The third application of this model, for which field validating information was obtainable, also involved a five-spot pilot site. An accidental increase in the sulfate concentration of injected leachant provided a chemical marker that was later measured in water samples taken at the recovery well. In this case, the model predicted breakthrough after 4 to 4½ days of pumping. A measurable increase in sulfate concentration at the recovery well occurred 4½ to 5 days following injection.

Site-Specific Model Revalidation and Calibration

Because the usefulness of this program as a predictive tool largely depends upon the accuracy of its assumptions concerning aquifer homogeneity, and since precise information on the type and extent of heterogeneities existing in a flow environment (faults, clay stringers, pinchouts) is generally inadequate, it is important that this program be revalidated in every aquifer environment where it is applied.

The use of nonreactive tracers, fluorescein dye, for example, is a preferred method of measuring breakthrough time for program validation. The problems involved in measuring change in pH can be avoided, and breakthrough time can be estimated with greater accuracy.

To some degree, empirically estimated user inputs such as permeability and the orientation of the permeability axis, porosity, ground water flow, and average aquifer thickness can be adjusted until model predictions of breakthrough time and drawdown correspond to field measurements of the same. This calibration process can be made as part of the initial pilot operation (generally a five-spot), and the program can then be used with greater confidence to model the leaching hydrology of a subsequent production phase conducted under similar aquifer conditions. Again, field measurements of breakthrough time at the recovery wells are useful for this calibration process.

The steady-state change in head (drawdown) that occurs at observation wells can provide another means of calibrating the model. The special point feature of the model can be used in this case to predict drawdown at the locations of observation wells.

The drawdown curve, developed as part of permeability measurement, is another useful calibrating tool which can be compared with the nonsteady or time-dependent drawdown predictions of the model. Here again, the coordinates of observation wells in the permeability test are input to the program as special points, along with the elapsed time for each head measurement, and a nonsteady flow simulation is generated.

USER OPTIONS AND INPUT REQUIREMENTS

The following user options are incorporated in the model: They delineate the scope of the program and indicate its applicability to individual field situations. A detailed description of ISL-50 input requirements is contained in appendix A.

Input Options

1. The user must specify whether the aquifer is leaky or nonleaky. A leaky aquifer is slightly permeable (permeability is assumed to be totally vertical) from an aquitard or semipervious confining layer above or below (but usually above) the confined aquifer.

2. The user must specify whether the flow is steady-state (time independent) or nonsteady (time dependent). Nonsteady simulation is applicable for the generally short time interval at the start of a pumping operation.

The time variable that enters into Δh calculation in nonsteady flow problems is not input by the user. Rather, new time values are generated by the program as the fluid front progresses along each streamline. This variable time dependence allows a more accurate simulation of front breakthrough time following startup or a change in pumping rates, which is critical when using front breakthrough time to validate the model.

3. The user may specify partially penetrating well screens. A three-dimensional flow simulation is developed for those cases in which the confined aquifer thickness is only partially screened (fig. 1).

4. Changes in pumping rates or pumping patterns at predetermined time points can be modeled in two ways. First, variable pumping patterns can be separated into two or more pumping patterns, each fixed throughout separate time intervals and each individually simulated. Second, the fluid front established in the first simulation can be recorded, and the altered paths of particles along this front can be traced under the next pumping pattern.

5. Permeability in this program is assumed to be either isotropic or of a directional nature. Horizontal major and minor axes of permeability may be specified if the hydrological field data so warrant. A vertical permeability can also be specified in partially penetrating screen situations.

6. The user may specify velocity and direction of ground water flow. Ground water flow is treated as a vector that can affect leachant velocity and direction.

7. Implicit in this description of modeling parameters is the user's selection of well locations. A maximum of 50 wells can be modeled. Locations and pumping rates may be varied to conform to site-specific conditions and to maximize uranium production and avoid contamination of the surrounding aquifer.

Output Options

The model output is presented both graphically and tabularly. Graphic displays of the fluid movement consist of three-dimensional streamline plots, isotime, isovelocity, and isopressure contour maps for the entire site or for portions of it. Tabular values are given for (1) pressure, velocity, and direction of particle flow at user-selected points, (2) volume of fluid passed between input and output wells and volume escaped into the aquifer (as time-dependent parameters), (3) front breakthrough time for each streamline developed, and (4) area swept by a well configuration (as a time-dependent parameter).

Principal Input Parameters

The following principal input parameters, most of which were treated above, are required by the program and are obtained from field or laboratory measurements: aquifer porosity, aquifer permeability, screen penetration, aquifer thickness, specific weight of leaching fluid, compressibility of fluid portion of the aquifer, input and output pumping rates, well coordinate locations, and time intervals during which the pumping and well pattern is constant.

PROGRAM STRUCTURE

A list of ISL-50 program subroutines and functions and a brief statement describing each component is contained in table 1.

The program flow of control is charted in figure 3, along with a brief description of the sequential steps involved.

TABLE 1. - Program Components

Routine	Purpose
INSITU.....	Execution control, output formatting.
INPT.....	Data input and initialization. Major and minor axes of permeability are specified.
DELTAH.....	Superposition of individual well solutions for head distribution.
VXVY.....	Calculation of flow velocity and direction, using head distribution and difference quotients. Ground water velocity is specified.
ODDPLOT.....	Detailed analysis of flow field at points of special interest to user.
STRM.....	Trace streamlines radiating from each recharge well by calculating a Taylor expansion about distance intervals for successive points in the flow path. Output streamline coordinates.
TMECONF.....	Analytic solution for head distribution in a time-dependent or nonsteady flow situation with confined nonleaky aquifer. Partial or total well screen penetration specified.
TMELKY.....	Analytic solution for head distribution in time-dependent flow with confined leaky aquifer. Partial or total well screen penetration specified.
SSLEAKY.....	Analytic solution for head distribution in steady-state flow with a confined aquifer. Partial or total well screen penetration specified.
SSCONF.....	Analytic solution for head distribution in steady-state flow with a confined nonleaky aquifer. Partial or total well screen penetration specified.
CNVRT.....	Axis rotation for alignment with major and minor permeability axis.
FILEDT.....	Edit output file for velocity and head contour map construction.
VFLOW.....	Determine the vertical flow component in cases of partially penetrating well screens.
Function	
BESSI.....	First-order I Bessel function approximation.
BESSK.....	First-order K Bessel function approximation.
WU.....	Well function approximation for nonleaky aquifer.
WU1.....	Well function approximation for leaky aquifer.

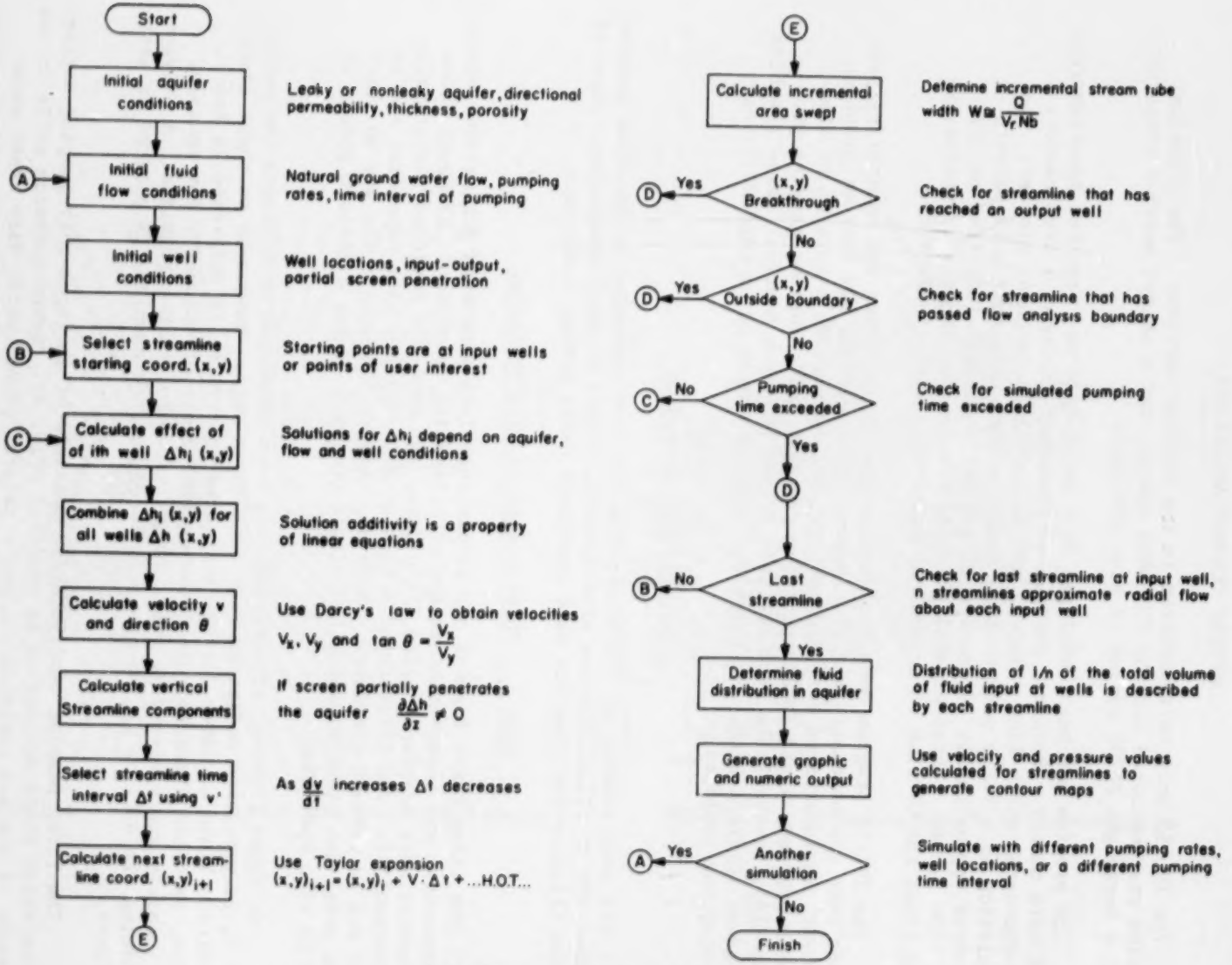
Start
Initial aquifer conditions
Leaky or nonleaky aquifer, directional permeability, thickness, porosity
A
Initial fluid flow conditions
Natural ground water flow, pumping rates, time interval of pumping
Initial well conditions
Well locations, input-output, partial screen penetration
B
Select streamline starting coord. (x,y)
Starting points are at input wells or points of user interest
C
Calculate effect of of ith well Δhi (x,y)
Solutions for Δhi depend on aquifer, flow and well conditions
Combine Δhi (x,y) for all wells Δh (x,y)
Solution additivity is a property of linear equations
Calculate velocity v and direction θ
Use Darcy's law to obtain velocities
Vx, Vy and tan θ = Vx/Vy
Calculate vertical Streamline components
If screen partially penetrates the aquifer ∂Δh/∂z ≠ 0
Select streamline time interval Δt using v'
As dv/dt increases Δt decreases
Calculate next streamline coord. (x,y)i+1
Use Taylor expansion
(x,y)i+1 = (x,y)i + V·Δt + ...H.O.T...
E
E
Calculate incremental area swept
Determine incremental stream tube width W ≅ Q/(Vr Nb)
D
Yes
(x,y) Breakthrough
Check for streamline that has reached an output well
No
D
Yes
(x,y) Outside boundary
Check for streamline that has passed flow analysis boundary
No
C
No
Pumping time exceeded
Check for simulated pumping time exceeded
Yes
D
B
No
Last streamline
Check for last streamline at input well, n streamlines approximate radial flow about each input well
Yes
Determine fluid distribution in aquifer
Distribution of 1/n of the total volume of fluid input at wells is described by each streamline
Generate graphic and numeric output
Use velocity and pressure values calculated for streamlines to generate contour maps
A
Yes
Another simulation
Simulate with different pumping rates, well locations, or a different pumping time interval
No
Finish

FIGURE 3. - Program flow of control.

COMPUTER PROGRAM ACCESSIBILITY

The ISL-50 program is written in two FORTRAN versions. The principal version was developed for the CDC 6600 system, and a second version compatible with a Burroughs B6700 system is also available.

The program has been developed in a batch mode and requires approximately 150K core memory. Required peripheral equipment includes a card reader, a 130-character-width line printer, and a digital incremental plotter. The simulation of a fully penetrating, steady-state, five-spot well pattern requires approximately 10 seconds of central processor (CP) time on the CDC 6600 system. Three-dimensional flow simulations require approximately five times the CP time of equivalent two-dimensional simulations.

The ISL-50 program does not incorporate the plotting and contouring programs that are necessary for development of streamline and contour graphics. However, an integrated software package, which includes a version of the ISL-50 program, a streamline plotting program, and a contour mapping program (16), is available from the Twin Cities Research Center. These graphics programs interact with, and therefore require access to, a CALCOMP library of plotting routines.

APPLICATIONS OF ISL-50

Six case studies of ISL-50 applications are presented to describe program output and to illustrate the site-specific nature of each simulation. Most of these illustrative graphics were generated using actual field data.

Case 1.--Three Stages of Site Operation

The first three figures are streamline plots generated during development, production, and restoration stages of a leaching operation. Although these graphics were developed using field data, the scenario depicted is partially hypothetical. The illustrated stages of activity are site dependent and so may not be present elsewhere. However, they do illustrate the way in which the model can provide useful hydrologic analysis during different phases of an in situ operation.

In these figures, streamlines have been developed to indicate the paths of fluid particles radiating from each input well. Hashmarks on these lines indicate fluid movement during equal time intervals, in this case 24 hours. Thus, particle traversal time between two points of interest, for example, a recharge and a discharge well, is indicated for each streamline. A polygon showing the outline of the ore pod, based on core samples, is plotted in each figure.

The first pumping configuration consists of an input-only pumping pattern from which fluid dispersion is assessed (fig. 4). Seven recharge wells in the center of the pod replaced the pod's natural ground water with fresh water. Certain ground water constituents were thought to inhibit the leaching process in this case. The ISL-50 program indicated the time required for freshwater to reach the periphery of the pod, about 7 days.

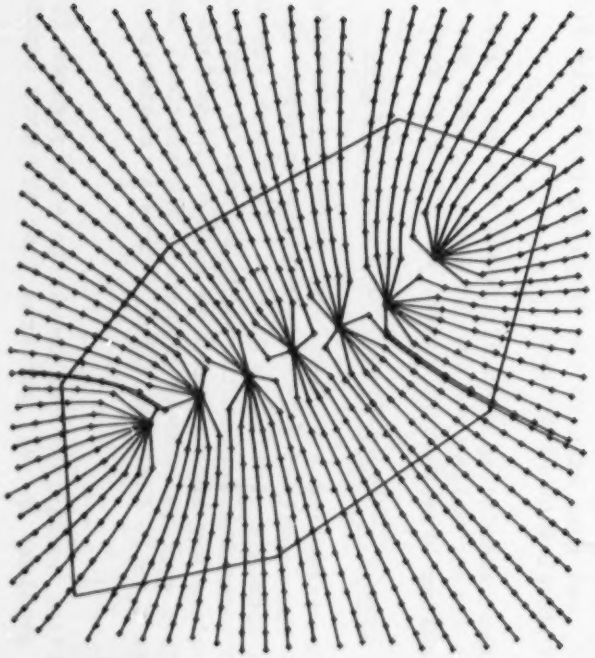

FIGURE 4. - Streamline pattern, site development stage.

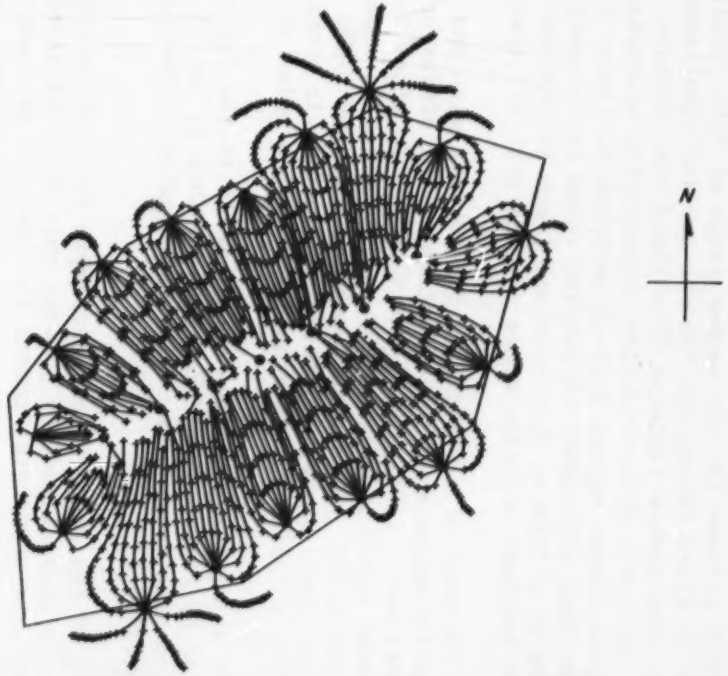

FIGURE 5. - Streamline pattern, operational stage.

The desired flow pattern for the second stage of this operation includes uniform dispersion of fluid throughout the ore pod and simultaneous maintenance of a more or less stable leachant flow field that extends lightly beyond the boundaries of the pod. This objective is partially accomplished in the balanced flow pattern (input equals output) depicted in figure 5. The streamline plot indicates that adjustment of pumping rates is necessary to prevent leachant excursions at points on the upper and lower boundaries of the pod and to insure that the leachant solution makes contact with all areas of the pod.

By comparing the area of a particular flow domain with the total area of an ore pod, a relative measure of the effectiveness of different pumping patterns can be obtained. In figure 5, the flow domain after 16 days of operation encompassed approximately 42,600 sq ft, while the approximate area of the pod is 49,300 sq ft. Adjustments must be made for fluid flow lines outside the pod, but approximately 85 pct of the pod is in contact with fluid, although the fluid's velocity varies considerably from point to point.

The effect of eliminating the two output wells at the far right of the pattern is depicted by figure 6. A plot such as this is useful in describing the fluid excursion that results if wells are inoperative because of plugging, or pump failure.

The beginning of the restoration process is depicted in figure 7. Ground water restoration is accomplished by pumping only from the seven wells in the center of the pod. The peripheral points in this plot are user-selected coordinates in the surrounding aquifer; they are chosen based on the information obtained from figure 5, which indicates the farthest extent of leachant flow during this stage. The information to be derived from figure 7 is the time required for fluid particles at these surrounding points to reach the output wells--in this case, approximately 17 or 18 days. Restoration is facilitated by minimizing the amount of leachant that enters the aquifer around the ore pod, and this adds to the importance of selecting an optimal pattern at earlier developmental stages.

Numerical output associated with figures 4 through 7 lists the amount of fluid reaching the output wells, the amount escaping into the surrounding aquifer, the streamline breakthrough times, and the area swept by each pattern.

Case 2.--Screen Versus Wellhead Locations

Figure 8 is a streamline plot developed using screen rather than wellhead locations. It is the result of a field situation in which the screens of a five-spot pattern were not vertically aligned with the wellheads. The precise location of the screens was determined using an onsite vertical deviation log. The wellhead locations are superimposed on this plot and form a symmetric pattern with 70-foot spacing between corners. The vertical deviations of screens range from 3 to 15 feet and clearly have a considerable effect on the flow field.

Unknown vertical deviations can result in even greater fluid flow prediction and control problems in well patterns that are more complex than this

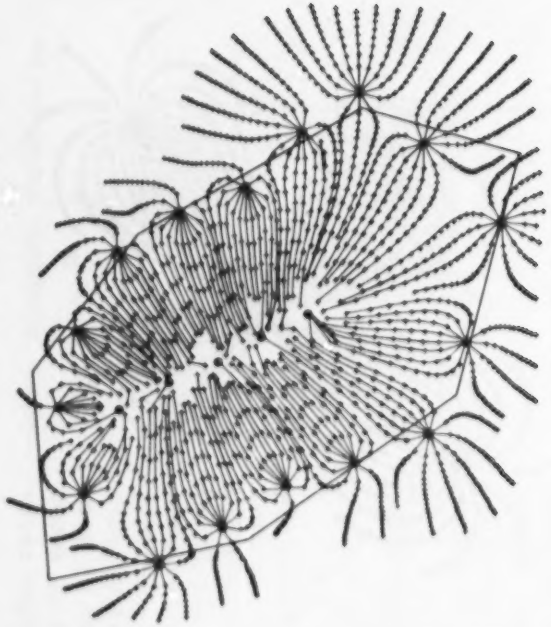

FIGURE 6. - Streamline pattern, fluid excursions.

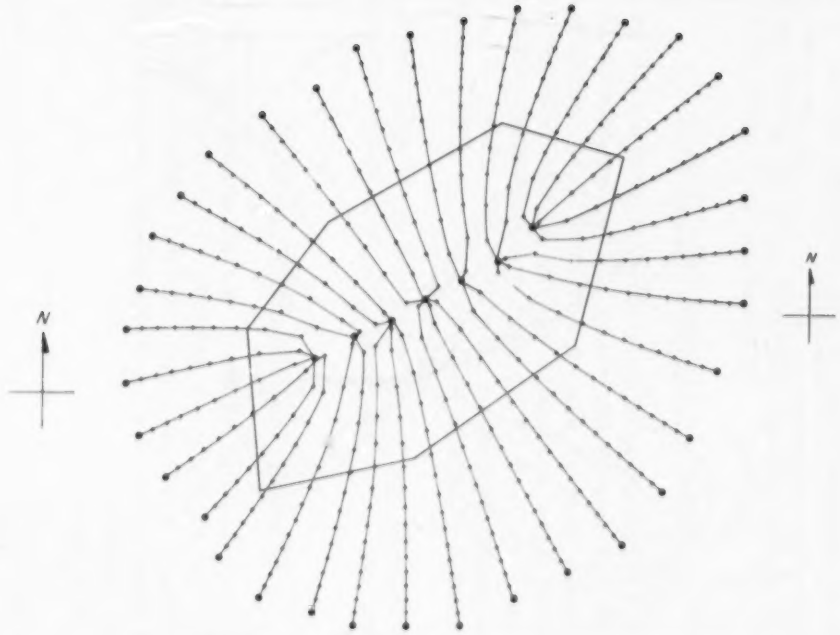

FIGURE 7. - Streamline pattern, restoration stage.

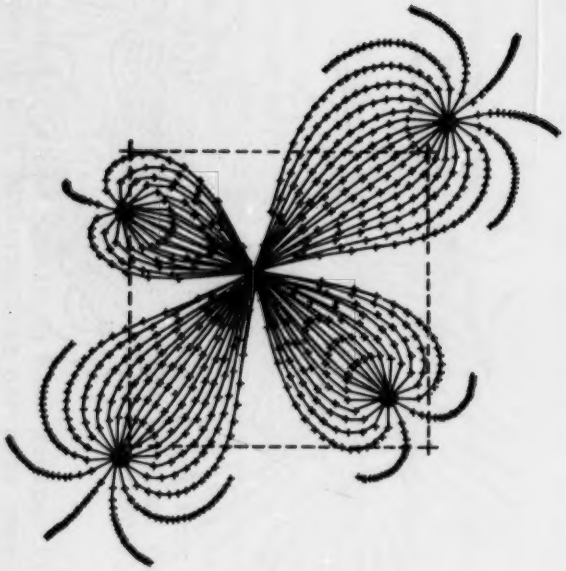

FIGURE 8. - Five-spot pattern using screen locations.

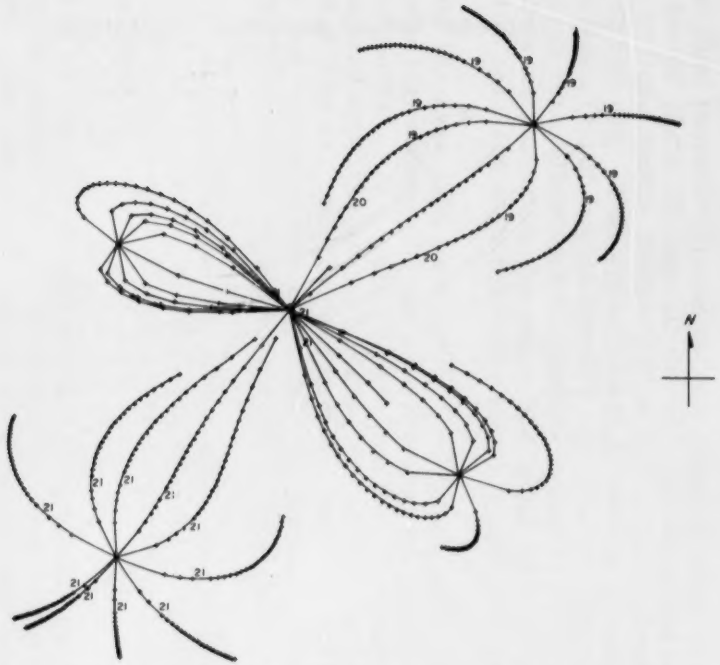

FIGURE 9. - Five-spot pattern with directional permeability.

five-spot pattern, as for example, the 23-well pattern of figure 5. It is apparent that a vertical deviation log is essential for control and optimization of fluid flow during in situ leaching and for accurate prediction by the ISL-50 program of in situ flow parameters.

Case 3.--Directional Anisotropy

The effect of directionally anisotropic permeability and natural ground water flow on a five-spot pattern is shown in figure 9. The major axis of permeability is oriented 163° from the x-axis in this case. The major permeability is an unusually high 6 darcys along the major axis, and 1 darcy along the minor axis. Ground water flow is 85 feet per year, in a direction 119° from the x-axis. This simulation is developed using screen, rather than wellhead, locations, which accounts for the slightly skewed well arrangement.

The ISL-50 input data deck for this hydrology simulation is included in appendix B. Appendix C contains, in condensed form, the tabular output generated by this run.

Case 4.--Flow Velocity Contours

In addition to streamline plots, isovelocity and isopressure contours can be developed using the ISL-50 program in conjunction with a contour mapping program. The relation between two hydrologic parameters--leachant velocity and flow direction--is illustrated by superimposing an isovelocity contour and a streamline plot, as shown in figure 10. The contour map in this figure is associated with an inverted five-spot pattern in which leachant is injected in the center well and recovered from four corner wells. Consequently, velocity is highest at the center of the pattern.

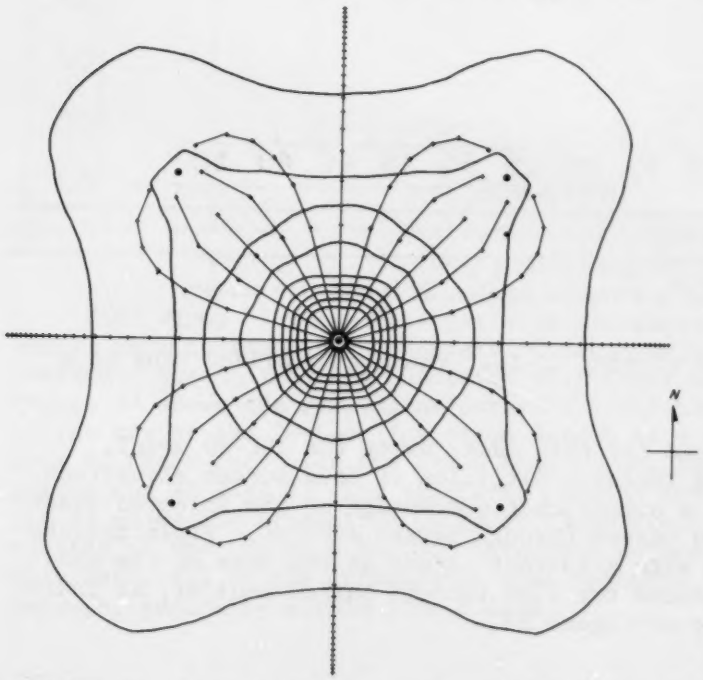

FIGURE 10. - Streamline pattern and isovelocity contour.

Case 5.--Three-Dimensional Simulation

The vertical component of leachant flow is of concern in situations in which the confined aquifer is only partially screened by injection and recovery wells. A partially screened condition, which is the rule for most in situ operations, is shown in figure 11. In this cross section, two 20-foot

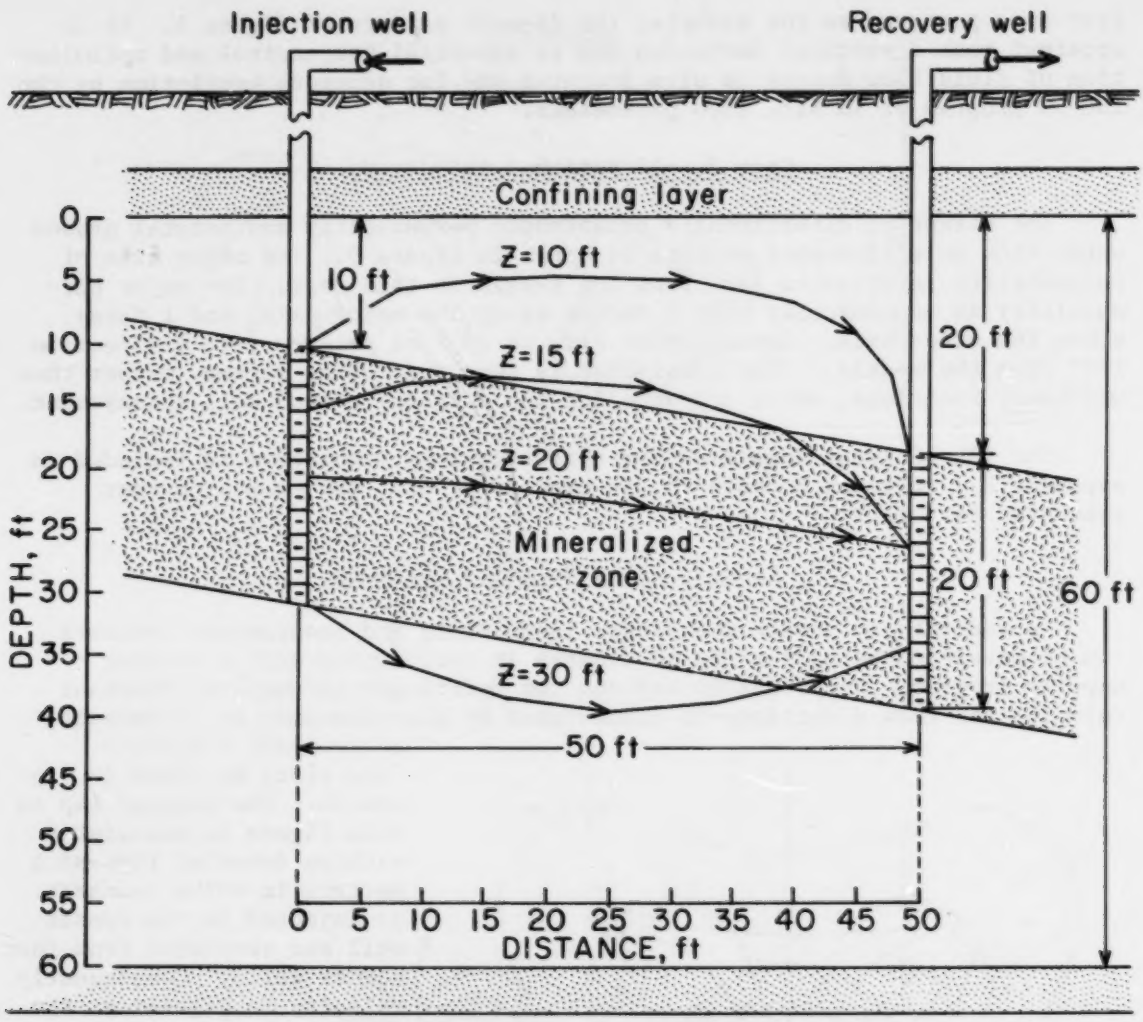

FIGURE 11. - Partially screened aquifer, 20-ft injection screen.

well screens are aligned with a sloping 20-foot-thick mineralized zone in a confined 60-foot aquifer.

Fluid streamlines developed for this case, using the ISL-50 model, describe the paths of injected leachant particles at four screen elevations. The streamlines indicate that a significant percentage of the injected leachant is flowing to the recovery screen through barren aquifer. Replacing the 20-foot injection well screen with a 10-foot screen at the base of the mineralized zone considerably reduces the flow through barren aquifer, as indicated by the simulated results of figure 12.

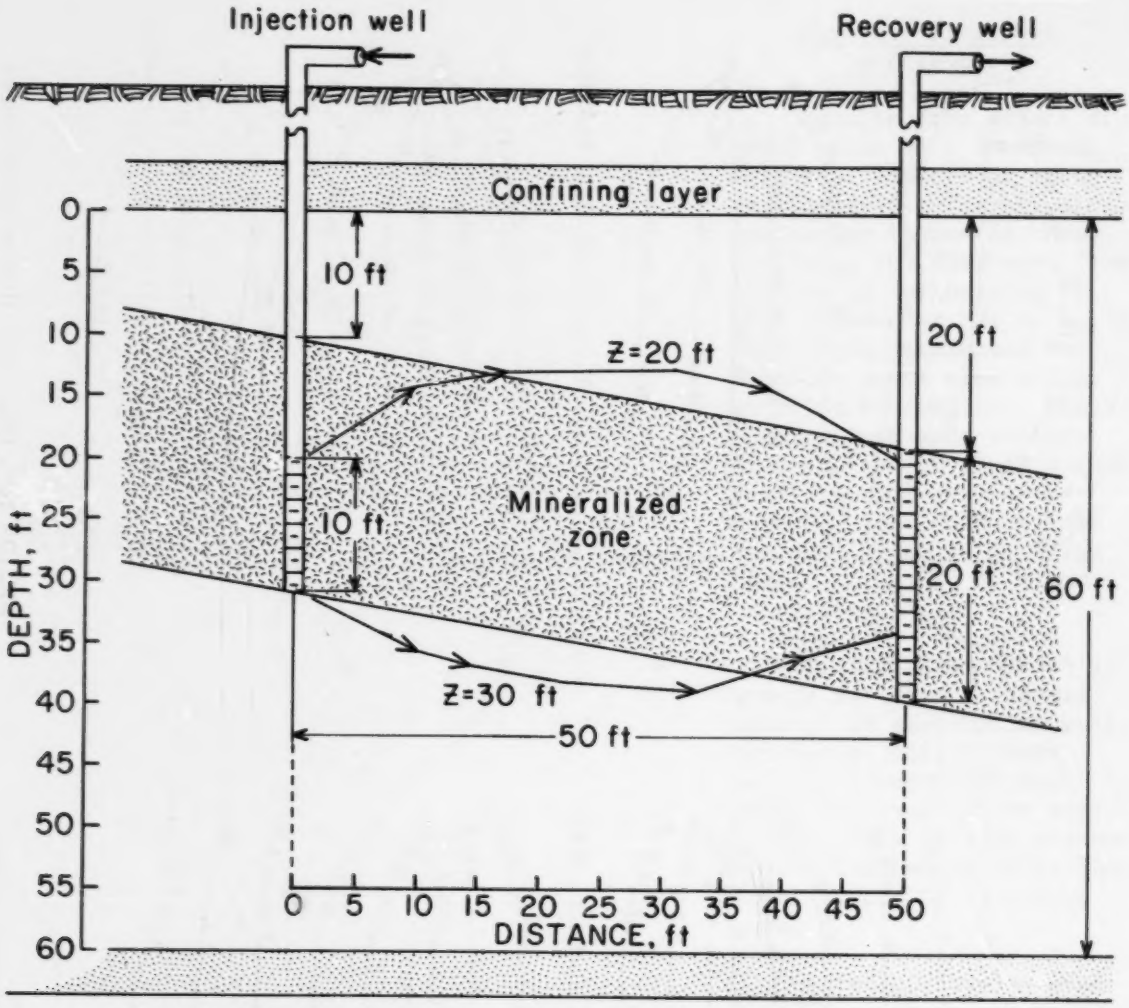

FIGURE 12. - Partially screened aquifer, 10-ft injection screen.

Plan views of these flow patterns are shown in figures 13 and 14. The entire well configuration consists of seven wells in a staggered-line drive pattern. Streamlines are annotated with their elevations in the aquifer. Figure 13 shows the streamline paths of fluid particles entering the aquifer at the top of the 20-foot injection screen (z = 10 in fig. 11). Note that none of these particles are drawn down to the level of the recovery screen during the 20 days of this simulated pumping operation. In contrast, figure 14 shows the streamline paths of those fluid particles entering the aquifer at the top of a 10-foot injection screen (z = 20 in fig. 12). In this case, all of the streamlines that approach the recovery screen break through, generally within 5 or 6 days.

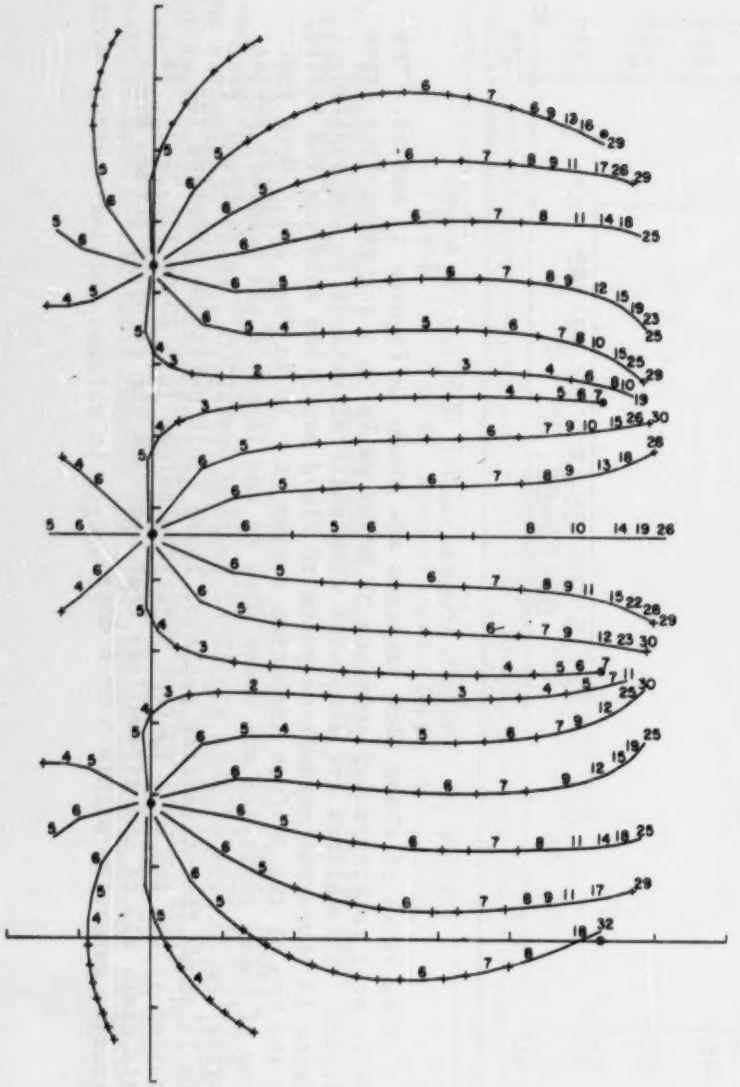

FIGURE 13. - Plan view, 20-ft injection screen.

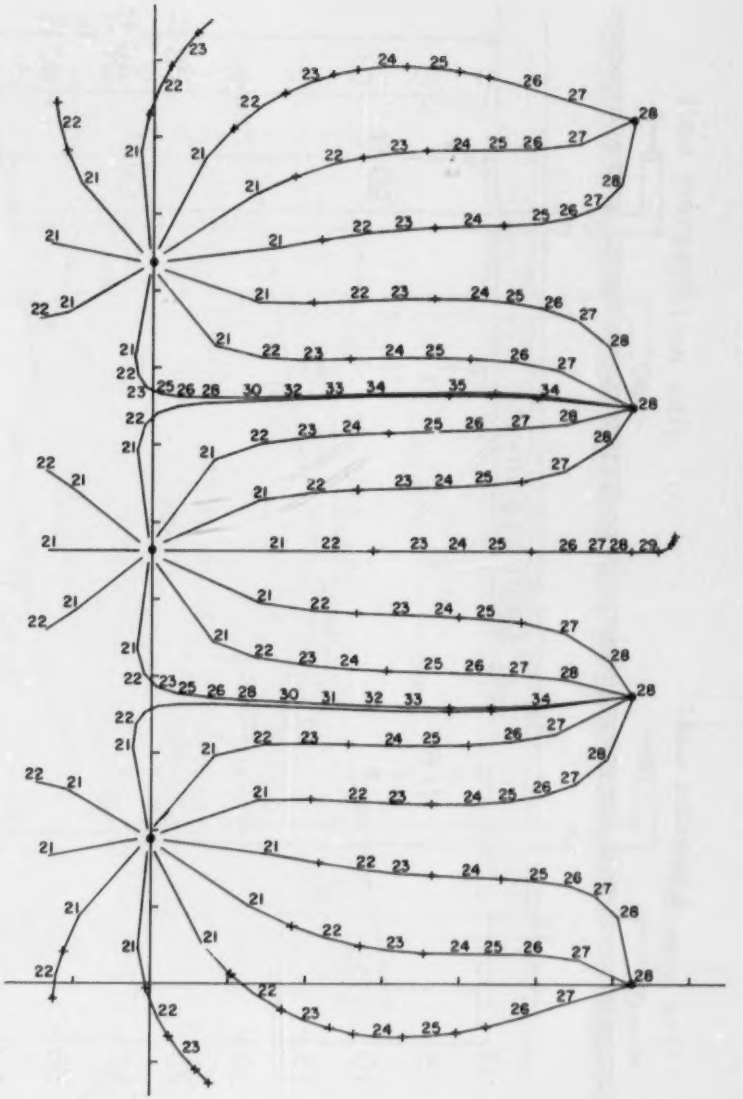

FIGURE 14. - Plan view, 10-ft injection screen.

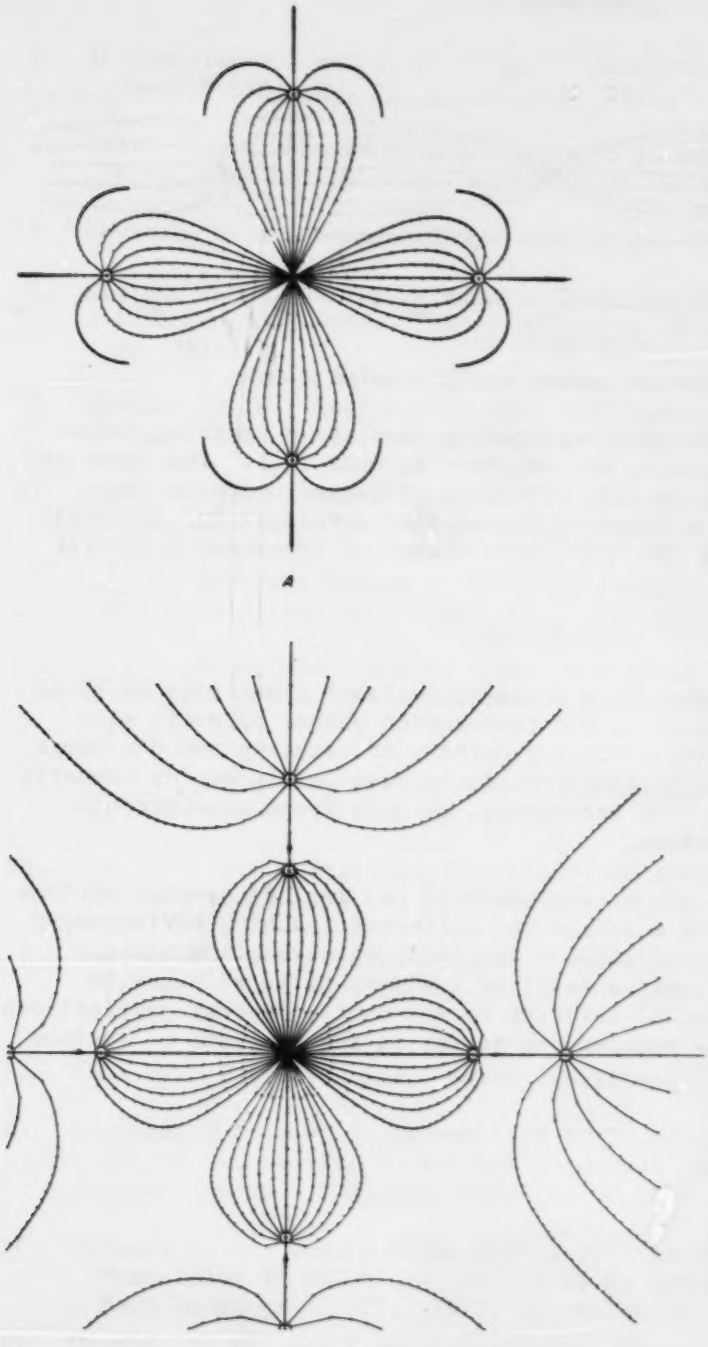

FIGURE 15. - Isolated five-spot and five-spot with four guard wells.

Case 6.--Five-Spot Confinement

Finally, the effect of guard wells on a symmetric five-spot pattern is shown in figure 15. The hydrology simulation indicates that confining the five-spot flow pattern by surrounding it with freshwater buffer wells selectively increases the velocity along slow-moving exterior streamlines. This results in a more uniform distribution of fluid within the five-spot and, consequently, in a more uniform leaching process, one less likely to result in fluid excursions.

Ground water, which is pumped in (from a distant source) at each buffer well, creates a local no-flow barrier between the four input well and buffer well pairs. This barrier concentrates leachant flow in the interior of the five-spot pattern.

After 30 days of simulated pumping from the isolated five-spot of figure 15A, 25 pct of injected leachant solution is still unrecovered by the center well. With the addition of four guard wells (figure 15B), unrecovered leachant after 30 days of pumping is less than 8 pct of total input.

The same image well technique can be applied to the modeling of geologic flow barriers if their locations are well established

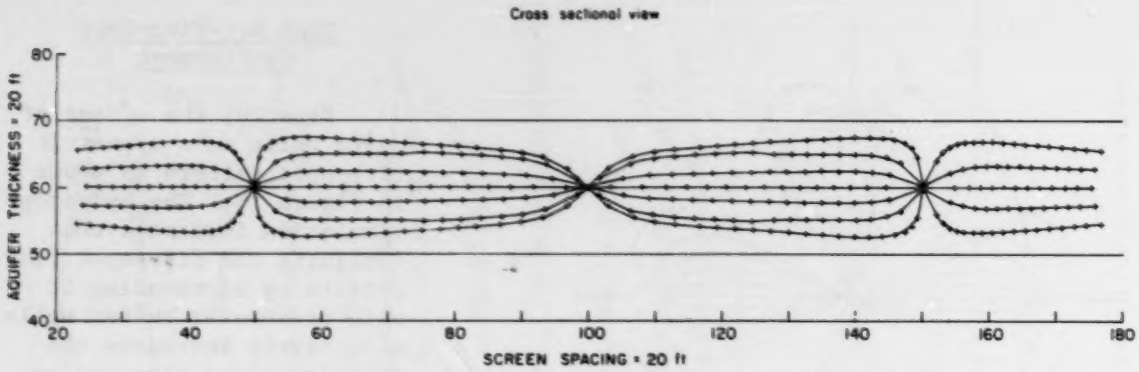

FIGURE 16. - Streamline pattern, aquifer-oriented screens.

through core data. Such a technique was used to simulate a leaching operation involving horizontal injection and recovery screens (13). The upper and lower aquifer confining layers in this case were simulated using an image well technique. Figure 16 is a cross sectional plot developed for the horizontal screen case and showing the hydrologic effect of impermeable aquifer boundaries.

CONCLUSIONS

The computer model described is a three-dimensional simulation of fluid flow during development, production, and restoration phases of an in situ leaching operation, involving an arbitrary pattern of recharge and discharge wells. Site-specific model validation procedures were developed for comparison of measured and predicted flow parameters, such as front breakthrough time and observation well drawdown.

Model input requirements are straightforward and include several options for modeling different types of aquifers and different leaching environments. Descriptive output is graphic and tabular and includes streamline plots, various isoparameter contours, and a detailed analysis of coordinates in the flow field that are of special interest to the user. Several applications were discussed to show how the program can assist in the analysis of various operational problems.

REFERENCES

1. Abramowitz, M., and I. A. Stegun. Handbook of Mathematical Functions. Dover Publishing, Inc., New York, 1964.

2. Craig, D. F. The Reservoir Engineering Aspects of Water Flooding. AIME, Soc. Petrol. Eng. Monograph, v. 3, New York, 1971, 135 pp.

3. DeWiest, R. J. Geohydrology. John Wiley & Sons, Inc., New York, 1965.

4. Glover, R. E., and W. T. Moody. Drawdown Due to Pumping in an Anisotropic Aquifer. Water Resources Bull., v. 12, October 1976, pp. 941-950.

5. Hantush, M. S. Analysis of Data From Pumping Tests in Leaky Aquifers. Trans. AGU, v. 37, December 1956, pp. 702-714.

6. _____. Aquifer Tests on Partially Penetrating Wells. J. Hydraulics Div., Proc. Am. Soc. Civil Eng., September 1961, pp. 171-195.

7. _____. Drawdown Around a Partially Penetrating Well. J. Hydraulics Div., Proc. Am. Soc. Civil Eng., July 1961, pp. 83-98.

8. _____. Non-Steady Flow to a Well Partially Penetrating an Infinite Leaky Aquifer. Proc. Iraqi Scientific Societies, v. 1, January 1957, pp. 10-19.

9. _____. Non-Steady Radial Flow in an Infinite Leaky Aquifer. Trans. AGU, v. 36, February 1955, pp. 95-100.

10. Jacob, C. E. Radial Flow in a Leaky Artesian Aquifer. Trans. AGU, v. 27, April 1946, pp. 198-208.

11. Kurth, D. I., and R. D. Schmidt. Computer Modeling of Five-Spot Well Pattern Fluid Flow During In Situ Uranium Leaching. BuMines RI 8287, 1978, 76 pp.

12. Lamb, H. Hydrodynamics. Dover Publishing, Inc., New York, 1945.

13. Morrell, R. J., W. C. Larson, and R. D. Schmidt (assigned to U.S. Dept. of the Interior). Horizontal Injection and Recovery Wells for In Situ Leach Mining. U.S. Pat. App. No. 60,101.

14. Murray, J. M. Ground Water Hydrology. Section in Handbook on the Principles of Hydrology, ed. by D. M. Gray. Water Information Center, Port Washington, N.Y., 1973, pp. 6.1-6.56.

15. Neuman, S. P., and P. A. Witherspoon. Theory of Flow in a Confined Two Aquifer System. Water Resources Res., v. 5, August 1969, pp. 803-816.

16. Tracy, F. T. A Computer Program for Contouring the Output of Finite Element Programs. U.S. Army Corps of Engineers Mis. Paper K-74-1, 1974, 26 pp.

APPENDIX A.--INPUT VARIABLE DESCRIPTION

Variable	Card columns	Description
		Flow card 1 (format I5)
ITOT	1-5	Total number of fluid flow analyses to be made.
AQR	6-10	Leaky or nonleaky aquifer parameter. If AQR = 1, aquifer is nonleaky; if AQR = 2, aquifer is leaky.
NUMODD	11-15	The number of special interest points, if any, that are to be evaluated.
NLINES	16-20	Number of fluid flow streamlines to be calculated. If NLINES = 0, streamline routine is omitted. However, a nonzero NLINES must be ≥ 3.
ITEREND	21-25	Maximum number of iterations allowed in Δh calculation. More iterations are required to evaluate points near the wells. Note: Erratic Δh values may indicate insufficient number of iterations.
NUMWL	26-30	Number of input and output wells (50 maximum).
MAXID	31-35	Maximum number of streamline coordinates that may be printed for any one streamline (MAXID $\leq$ 300).
DISTIME	36-40	Time interval between printed streamline points, in hours (DISTIME ≥ 1).
		Flow card 2 (format F10.0)
B	1-10	Thickness of aquifer being leached, in feet.
Z	11-20	The starting z-coordinate of streamlines. WELL (1,i) $\leq$ z $\leq$ WELL (2,i) for all injection wells, in feet.
RADIUS	21-30	Well radius, in inches.
GAMMA	31-40	Specific weight of leaching fluid, in lb/cu ft.
TERM	41-50	Termination value for convergent series calculating DELPART(i), the influence of individual wells on each matrix point (range 1×10^{-5} to 1×10^{-1}). Points very near the well require more iterations for stable values.
		Flow card 3 (format E15.5)
ALPHA	1-15	Compressibility of aquifer solid, in sq ft/lb.
BETA	16-30	Compressibility of fluid, in sq ft/lb.

Variable	Card columns	Description
Permeability/porosity card (format F10.0)		
N	1-10	Porosity of aquifer, void volume/total volume, dimensionless.
KMJR	11-20	Aquifer permeability along the major horizontal axis of permeability, darcys. (1 darcy ≈ 3.156 X 10^{-7} ft/sec)
DMJR	21-30	Orientation of the major permeability axis--the angle from the positive x-axis to the KMJR axis, in degrees ($0 \leq$ DMJR ≤ 360).
KMNR	31-40	Aquifer permeability along the minor permeability axis (minor axis is orthogonal to DMJR), darcys.
KZ	41-50	Aquifer permeability along the vertical (z) axis, darcys.
Boundary card (format F10.0)		
XMIN XMAX YMIN YMAX	1-10 11-20 21-30 31-40	Horizontal boundaries of hydrology analysis, in feet.
Ground water card (format F10.0)		
WATDR	1-10	Direction of ground water flow. The angle from the positive x-axis to the positive direction of ground water flow, in degrees ($0 \leq$ WATDR ≤ 360).
WATRD	11-20	Ground water velocity, in feet per year.
One well card is required for each discharge or recharge well in the pattern (number of well cards = NUMWL).		
Well card(s) (format F10.0)		
WELL (1,i)	1-10	The distance from the top of the confined aquifer to the top of well screen i, in feet.
WELL (2,i)	11-20	The distance from the top of the confined aquifer to the bottom of well screen i, in feet.
WELL (3,i)	21-30	x-coordinate of well, in feet.
WELL (4,i)	31-40	y-coordinate of well, in feet.
WELL (5,i)	41-50	Well pumping start time, in hours. Note the constraint on the subsequent time card(s).
WELL (6,i)	51-60	Well pumping stop time, in hours.
WELL (7,i)	61-70	Well flow rate in gal/min. If well (7,i) < 0, it is an injection well; If well (7,i) > 0, it is a recovery well.
WELL (8,i)	71-80	Well number.

One time card is required for each separate hydrology analysis. If no wells are operating at TIME(j), the analysis is skipped.

(Number of time cards = ITOT)

Time card(s)			
Variable	Format	Card columns	Description
TIME(j)	F10.0	1-10	The time interval after which the jth hydrology analysis is made, in hours. Each well must be continuously operative or inoperative during the entire time interval. TIME (j) - TIME (j-1) (TIME(j) > TIME (j-1)).
ITME(j)	I10	11-20	Time-dependent or time-independent flow parameter. If ITME(j) = 1, flow is assumed to have reached a steady state. If ITME(j) = 2, flow is time dependent. In this case, flow analysis is usually made for the normally short time interval prior to steady state.

The following card is required in the leaky aquifer case (AQR > 1).

Leaky aquifer card (format F10.0)		
BPRIME	1-10	The thickness of the aquifer above the aquifer to be leached, in feet.
KPRIME	11-20	The permeability of the aquifer above the aquifer to be leached, darcys.

One oddpoint card is required for each special interest point at which a separate description of fluid flow behavior is desired. A streamline beginning at this point is also developed (number of oddpoint cards = NUMODD).

Oddpoint card(s) (format F10.0)		
ODDPT(3,i)	21-30	x-coordinate of special interest point, in feet.
ODDPT(4,i)	31-40	y-coordinate of special interest point, in feet.
ODDPT(8,i)	71-80	Point number.

APPENDIX B.--SAMPLE INPUT DATA

```
    1     1     1    15  300     5  300    12     (Flow Card 1)
34.0          20.0       4.0        62.4        .00001   (Flow Card 2)
        2.57000E-8       2.32000E-8                      (Flow Card 3)
.28           1.56       23.        .62         .98      (Permeability/Porosity Card)
-50.0         0.0        -10.0      20.0                 (Boundary Card)
7.0           14.0                                       (Ground water Card)
13.           26.0       -31.7      12.0        0.0        1000.      -4.95      16.
15.0          31.0       45.2       34.7        0.0        1000.      -4.95      17.0
15.0          24.0       31.8       -30.6       0.0        1000.      -4.95      18.    (Well Cards)
10.0          22.0       -31.7      -46.2       0.0        1000.      -4.95      19.
18.0          27.0       0.0        0.0         0.0        1000.      21.9       15.
999.                   1                                  (Time Card)
                         -29.0      2.5                                          6.0 (Odd point Card)
```

APPENDIX C.--SAMPLE OUTPUT

LEACHANT FLOW SIMULATION

SIMULATION NUMBER 1 CONDUCTED FOR 499.0 HOURS. AT 499.0 HOURS AFTER START OF PUMPING OPERATION.

AQUIFER IS NON-LEAKY

HEAD CHANGE IS TIME INDEPENDENT

AQUIFER POROSITY = .30000000

AQUIFER PERMEABILITY
(MAJOR AXIS) = 2.0000000 DARCYS
(MINOR AXIS) = 2.0000000 DARCYS
VERTICAL = 2.0000000 DARCYS

GROUNDWATER VELOCITY = 0.0 FT./YR.
DIRECTION = 0.00 DEGREES (FROM POS. X-AXIS)

WELL RADIUS 4.0 INS.

TIME INTERVAL OF PARTICLE FLOW EVALUATION 24 HRS.

AQUIFER THICKNESS = 34.00 FT.

DEPTH OF FLOW ANALYSIS (Z) = 20.00 FT.

BOUNDARIES OF AQUIFER ANALYSIS (FT.)

X-MAX	X-MIN	Y-MAX	Y-MIN
100.0	-85.0	85.0	-100.0

WELLS OPERATING AT TIME = 499.0 HOURS.

WELL NO.	LOCATION(FT.) X-COORD	Y-COORD	PUMPING RATE(GAL/MIN)	SCREEN LOCATION(FT.) TOP	BOTTOM	WELL START TIME(HRS.)	WELL STOP TIME(HRS.)
15	-.0	-.0	15.00 INPUT	0.00	34.00	0.0	500.0
15	50.0	-.0	15.00 OUTPUT	0.00	34.00	0.0	500.0

SPECIAL POINTS TO BE EVALUATED

POINT NO.	LOCATION(FT.) X-COORD	Y-COORD
1	3.0	-.0
2	25.0	-.0

EVALUATION OF SPECIAL INTEREST POINTS

X CO-ORD (FEET)	Y CO-ORD (FEET)	DRAWDOWN (FEET)	VELOCITY (FEET/SEC.)	DIRECTION OF FLOW (DEGREES)
.30000E+01	-.45475E-12	-.67916E+01	.18492E-03	0.
.25000E+02	-.31032E-11	-.84122E-13	.41718E-04	0.

STREAMLINE NUMBER 4 BEGINNING AT WELL (OR SPECIAL POINT) NUMBER 15

INITIAL ANGLE OF THIS STREAMLINE = 0.00 DEGREES. (IN HORIZONTAL PLANE)

NUMBER OF COORDINATES PRODUCED = 11

STREAMLINE FRONT HAS BROKEN THROUGH AT WELL NUMBER 15 AFTER 221.27 HOURS OF WELL OPERATI

X CO-ORD	1.33	10.71	15.09	19.11	22.42	26.44	30.12	34.05	38.61	45.42	58.88
Y CO-ORD	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00
Z CO-ORD	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00	20.00

TWO DIMENSIONAL AREA SWEPT BY STREAMLINE NO. 4 = 634.76 SQ.FT.

BREAKDOWN OF FLUID DISPERSION FROM INPUT WELL NO. 15 AFTER 499.0 HRS. OF WELL OPERATION

FLUID INPUT AT WELL NO. 15 OUTPUT AT WELL NO.
15 .42548E+05 GAL.

PERCENT OF WELL NO. 15 FLUID WITHDRAWN FROM AQUIFER 13.9 PERCENT
PERCENT OF WELL NO. 15 FLUID NOT WITHDRAWN FROM AQUIFER 86.1 PERCENT

TWO DIMENSIONAL AREA SWEPT BY WELL NO. 15 = 5153.31 SQ.FT.

TOTAL TWO DIMENSIONAL AREA SWEPT BY THIS WELL CONFIGURATION = 5153.31 SQ.FT.

100 TOTAL STREAMLINE COORDINATES OUTPUT TO TAPE 4.
THE FINAL 4 GRIDDED POINTS ARE WELL COORDINATES AND SPECIAL INTEREST POINTS.

100 HEAD AND VELOCITY VALUES FOR CONTOUR MAP CONSTRUCTION OUTPUT TO TAPE 12.

APPENDIX D.--ISL-50 PROGRAM LISTING

```
C*****INSI  10
C*****INSI  20
C*****INSI  30
C                                                                       INSI  40
C                    U.S.B.M. IN-SITU LEACHING HYDROLOGY MODEL           INSI  50
C                                                                       INSI  60
C                                ISL-50                                 INSI  70
C                                                                       INSI  80
C            DEVELOPED AND PROGRAMMED AT T.C.M.R.C. BY: R.D. SCHMIDT    INSI  90
C                                                                       INSI 100
C                          (LATEST MOD. APR. 1979)                      INSI 110
C                                                                       INSI 120
C*****INSI 130
C*****INSI 140
C*****INSI 150
      PROGRAM  INSITU (INPUT, OUTPUT, TAPE5=INPUT, TAPE12, TAPE6,       INSI 160
     1   TAPE13, TAPE14, TAPE15, TAPE7)                                 INSI 170
C                                                                       INSI 180
C***** THIS PROGRAM EVALUATES PARTICLE FLOW PARAMETERS ASSOCIATED        INSI 190
C***** WITH INSITU LEACHING ACTIVITY.  THE PROGRAM WILL ACCOMODATE A     INSI 200
C***** LEACHING OPERATION OF UP TO 50 WELLS.  A LARGER WELL PATTERN      INSI 210
C***** CAN BE SIMULATED THROUGH MANIPULATION OF COMMON STORAGE .         INSI 220
C***** RESULTS FROM THIS PROGRAM SHOWING PARTICLE STREAMLINES CAN BE     INSI 230
C***** OUTPUT TO ONLINE DISC STORAGE FOR LATER INPUT TO A PLOTTING       INSI 240
C***** PROGRAM.                                                         INSI 250
C                                                                       INSI 260
      COMMON  /CROSS/  NCSPT, CSTRM (3000, 5), LFLAG                    INSI 270
      COMMON  /PIE/  PI                                                 INSI 280
      COMMON  /WATER/  WATRX, WATRY, WATDR, WATRV                       INSI 290
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME        INSI 300
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN              INSI 310
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,   INSI 320
     1   K1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM               INSI 330
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ               INSI 340
      COMMON  /LINES/  OUTOPT, MAXID, DISTME, CUTOFF                    INSI 350
      DIMENSION   ANGL1 (50), CTLPT (2,100), ITME (10), ODDLIST (9,50),INSI 360
     1   ODDPT (9,50), OUTA (2), OUTB (2), OUTC (2), OUTD (2), OUTE (2),INSI 370
     2   TIME (10), WELL (9,50)                                         INSI 380
      INTEGER      AQR, DISTME, HRS, OUTOPT                             INSI 390
      REAL         K, KMJR, KMNR, KPRIME, KX, KY, KZ, K1, N, N1         INSI 400
      DATA    OUTA, OUTB, OUTC, OUTD, OUTE / 8HINDEPEND, 8HDEPENDEN,    INSI 410
     1   8HENT      , 8HT        , 8HNON-LEAK , 8HLEAKY    , 8HY       ,INSI 420
     2   8H         , 8HINPUT    , 8HOUTPUT    /                       INSI 430
      PRINT   10                                                        INSI 440
C*****INSI 450
      LIST  =  0                                                        INSI 460
      PI  =  3.141592653589793                                          INSI 470
 1000 CALL INPT (NUMODD, ODDPT, CTLPT, START)                           INSI 480
      NUMTOT  =  NUMODD + NUMWL                                         INSI 490
      SIZE  =  AMAX1 (XMAX - XMIN, YMAX - YMIN)                         INSI 500
 1010   DO 1030  I  =  1, NUMTOT                                        INSI 510
        CTLPT (1, I)  =   (CTLPT (1, I) - XMIN) / SIZE * 1000.          INSI 520
```

```
        CTLPT (2, I)  =   (CTLPT (2, I) - YMIN) / SIZE * 1000.          INSI 530
        IF (I .GT. NUMWL)  GO TO 1020                                   INSI 540
        WELL (3, I)  =  CTLPT (1, I)                                    INSI 550
        WELL (4, I)  =  CTLPT (2, I)                                    INSI 560
        WELL (7, I)  =  WELL (7, I) * (1000. / SIZE) *  * 3             INSI 570
        WELL (2, I)  =  WELL (2, I) / SIZE * 1000.                      INSI 580
        WELL (1, I)  =  WELL (1, I) / SIZE * 1000.                      INSI 590
        GO TO 1030                                                      INSI 600
1020    ODDPT (3, I - NUMWL)  =  CTLPT (1, I)                           INSI 610
        ODDPT (4, I - NUMWL)  =  CTLPT (2, I)                           INSI 620
1030    CONTINUE                                                        INSI 630
      THICK  =  B                                                       INSI 640
      RAD  =  RADIUS                                                    INSI 650
      HRS  =  DISTME / 3600.                                            INSI 660
      RADIUS  =  RADIUS / (12. * SIZE) * 1000.0                         INSI 670
      B  =  B / SIZE * 1000.                                            INSI 680
      GAMMA  =  GAMMA * (SIZE / 1000.) *  * 3                           INSI 690
      KX  =  KX / SIZE * 1000.                                          INSI 700
      KY  =  KY / SIZE * 1000.                                          INSI 710
      KZ  =  KZ / SIZE * 1000.                                          INSI 720
      ALPHA  =  ALPHA * (1000. / SIZE) *  * 2                           INSI 730
      BETA  =  BETA * (1000. / SIZE) *  * 2                             INSI 740
      DEPTH  =  Z                                                       INSI 750
      Z  =  SQRT (KX /.KZ) * Z / SIZE * 1000.                           INSI 760
      BPRIME  =  BPRIME / SIZE * 1000.                                  INSI 770
      KPRIME  =  KPRIME / SIZE * 1000.                                  INSI 780
      LFLAG  =  0                                                       INSI 790
      WATDD  =  WATDR                                                   INSI 800
        DO 1040  II  =  1, NUMWL                                        INSI 810
        IF (WELL(2,II)-WELL(1,II) .LT. B)  GO TO 1050                   INSI 820
1040    CONTINUE                                                        INSI 830
      GO TO 1060                                                        INSI 840
1050  LFLAG  =  1                                                       INSI 850
1060    DO 1230  IRUN  =  1, ITOT                                       INSI 860
        L1  =  0                                                        INSI 870
        REWIND 13                                                       INSI 880
        T1  =  TIME (IRUN) / 3600.                                      INSI 890
        T2  =  T1                                                       INSI 900
        IF  (IRUN .GT. 1)  T2  =   (TIME (IRUN) - TIME (IRUN - 1)) /    INSI 910
     1    3600.                                                        INSI 920
C***** PRINT HEADINGS AND INPUT VALUES.                                  INSI 930
        PRINT   20, IRUN, T2, T1, OUTC (AQR), OUTD (AQR), RAD, OUTA     INSI 940
     1    (ITME(IRUN)), OUTB (ITME (IRUN)), HRS, N, THICK, DEPTH, KMJR,INSI 950
     2    KMNR, ZK, WATRV, XMAX, XMIN, YMAX, YMIN, WATDR               INSI 960
      WATDR  =  WATDR * PI / 180. + DMJR                                INSI 970
      IF  (WATDR .GE. 2 * PI)  WATDR  =  WATDR - 2 * PI                 INSI 980
      IF  (WATDR .LT. 0.0)  WATDR  =  WATDR + 2. * PI                   INSI 990
      WATRV  =  WATRV * (1000. / SIZE)                                  INSI1000
      WATRX  =  COS (WATDR) * WATRV / 3.1536E7                          INSI1010
      WATRY  =  SIN (WATDR) * WATRV / 3.1536E7                          INSI1020
      DMJR  =  DMJR * PI / 180.                                         INSI1030
      DMNR  =  DMJR + PI / 2.0                                          INSI1040
```

```
      PRINT   30, T1                                                    INSI1050
        DO 1090  I  =  1, NUMWL                                         INSI1060
        IF (WELL(5,I) .GE. TIME(IRUN) .OR. WELL(6,I) .LT. TIME(IRUN))INSI1070
     1    GO TO 1070                                                   INSI1080
        IGAL  =  2                                                     INSI1090
        GAL  =  ABS (WELL(7, I) * 7.4805 * 60. * ((SIZE / 1000.) *  *INSI1100
     1    3))                                                          INSI1110
        IF  (WELL(7, I) .LT. 0)  IGAL  =  1                            INSI1120
        W3  =  WELL (3, I) / 1000. * SIZE + XMIN                       INSI1130
        W4  =  WELL (4, I) / 1000. * SIZE + YMIN                       INSI1140
        W2  =  WELL (2, I) / 1000. * SIZE                              INSI1150
        W1  =  WELL (1, I) / 1000. * SIZE                              INSI1160
        IW7  =  WELL (8, I)                                            INSI1170
        RUNO  =  WELL (5, I)                                           INSI1180
        W5  =  WELL (5, I) / 3600.                                     INSI1190
        W6  =  WELL (6, I) / 3600.                                     INSI1200
        PRINT   40, IW7, W3, W4, GAL, OUTE (IGAL), W1, W2, W5, W6      INSI1210
1070    CONTINUE                                                       INSI1220
        CALL CNVRT (WELL(3, I), WELL(4, I), DMJR)                      INSI1230
1080    CONTINUE                                                       INSI1240
      IF (NUMODD .LE. 0)  GO TO 1100                                   INSI1250
      PRINT   50                                                       INSI1260
        DO 1090  I  =  1, NUMODD                                       INSI1270
        P3  =  ODDPT (3, I) / 1000. * SIZE + XMIN                      INSI1280
        P4  =  ODDPT (4, I) / 1000. * SIZE + YMIN                      INSI1290
        IP7  =  ODDPT (8, I)                                           INSI1300
        PRINT   60, IP7, P3, P4                                        INSI1310
        CALL CNVRT (ODDPT(3, I), ODDPT(4, I), DMJR)                    INSI1320
1090    CONTINUE                                                       INSI1330
1100  DMJR  =   - DMJR                                                INSI1340
      IF  (IRUN .GT. 1)  RUNO  =  TIME (IRUN - 1)                      INSI1350
C*********************************************************************INSI1360
C***** INITIATE COMPUTATION OF ODDPOINTS, IF REQUIRED.                 INSI1370
C***** ODDPOINT ROUTINE WILL INITIATE STREAMLINE CALCULATION           INSI1380
C***** AT THIS POINT.                                                 INSI1390
1110  IF (NUMODD .LE. 0)  GO TO 1140                                   INSI1400
      PTME  =  TIME (IRUN) - RUNO                                      INSI1410
      CALL ODDPLOT (PTME, NUMODD, ODDLIST, ODDPT, IRUN)                INSI1420
      PRINT   10                                                       INSI1430
      PRINT   70                                                       INSI1440
        DO 1120  NODD  =  1, NUMODD                                    INSI1450
        ODDPT (3, NODD)  =  ODDPT (3, NODD) - (RADIUS + 1) * COS       INSI1460
     1    (ODDLIST (9, NODD) * PI / 180.)                              INSI1470
        ODDPT (4, NODD)  =  ODDPT (4, NODD) - (RADIUS + 1) * SIN       INSI1480
     1    (ODDLIST (9, NODD) * PI / 180.)                              INSI1490
        ODDPT (5, NODD)  =  0.0                                        INSI1500
        ODDPT (6, NODD)  =  TIME (IRUN) + 1.0                          INSI1510
        CALL CNVRT (ODDLIST(3, NODD), ODDLIST(4, NODD), DMJR)          INSI1520
        ODDLIST (3, NODD)  =  ODDLIST (3, NODD) * SIZE / 1000. + XMININSI1530
        ODDLIST (4, NODD)  =  ODDLIST (4, NODD) * SIZE / 1000. + YMININSI1540
        ODDLIST (5, NODD)  =  ODDLIST (5, NODD) * SIZE / 1000.         INSI1550
        ODDLIST (8, NODD)  =  ODDLIST (8, NODD) * SIZE / 1000.         INSI1560
```

```
            ODDPT (7, NODD)  =   - ODDLIST (8, NODD) * PI * RADIUS *  * 2INSI1570
            ANGL1 (NODD)  =  ODDLIST (9, NODD)                          INSI1580
            ODDLIST (9, NODD)  =  ODDLIST (9, NODD) - DMJR * 180. / PI  INSI1590
            PRINT   80, (ODDLIST(J, NODD), J=3, 5), (ODDLIST (J, NODD),  INSI1600
     1        J=8, 9)                                                   INSI1610
 1120       CONTINUE                                                    INSI1620
 1130     CONTINUE                                                      INSI1630
C***** CALCULATE STREAMLINES FOR INCIDENTAL POINTS. (NOT WELLS)         INSI1640
         CALL STRMLNE (N, L1, ODDPT, NUMODD, RADIUS, 1, ANGL1, TIME,    INSI1650
     1     IRUN, WELL, NUMWL, LIST, Z)                                 INSI1660
         IF (NLINES .LE. 0)  GO TO 1160                                INSI1670
 1140      DO 1150  I  =  1, NLINES                                     INSI1680
 1150      ANGL1 (I)  =  360.0 / NLINES                                 INSI1690
C***** INITIATE CALCULATION OF STREAMLINES,IF REQUIRED.                 INSI1700
         CALL STRMLNE (N, L1, WELL, NUMWL, RADIUS, NLINES, ANGL1, TIME, INSI1710
     1     IRUN, WELL, NUMWL, LIST, Z)                                 INSI1720
 1160    CONTINUE                                                      INSI1730
         NSUB  =  0                                                    INSI1740
         IF (NUMODD .LE. 0)  GO TO 1180                                INSI1750
           DO 1170  I  =  1, NUMODD                                    INSI1760
           ODDPT (3, I)  =  CTLPT (1, NUMWL + I)                       INSI1770
           ODDPT (4, I)  =  CTLPT (2, NUMWL + I)                       INSI1780
 1170      CONTINUE                                                    INSI1790
 1180    CONTINUE                                                      INSI1800
           DO 1210  I  =  1, NUMTOT                                    INSI1810
           C1  =  CTLPT (1, I) * SIZE / 1000. + XMIN                   INSI1820
           C2  =  CTLPT (2, I) * SIZE / 1000. + YMIN                   INSI1830
           IF (C1 .GE. XMIN .AND. C1 .LE. XMAX)  GO TO 1190            INSI1840
           NSUB  =  NSUB + 1                                           INSI1850
           GO TO 1210                                                  INSI1860
 1190      IF (C2 .GE. YMIN .AND. C2 .LE. YMAX)  GO TO 1200            INSI1870
           NSUB  =  NSUB + 1                                           INSI1880
           GO TO 1210                                                  INSI1890
 1200      WRITE (6,   90)  C1, C2                                     INSI1900
           WRITE (13,   90)  C1, C2                                    INSI1910
 1210      CONTINUE                                                    INSI1920
         L1  =  L1 + NUMTOT - NSUB                                     INSI1930
         TOTA  =  0.0                                                  INSI1940
           DO 1220  I  =  1, NUMWL                                     INSI1950
           CALL CNVRT (WELL(3, I), WELL(4, I), DMJR)                   INSI1960
           IF (WELL(7,I) .GE. 0.0)  GO TO 1220                         INSI1970
           TOTA  =  TOTA + WELL (9, I)                                 INSI1980
 1220      CONTINUE                                                    INSI1990
         WATDR  =  WATDD                                               INSI2000
         DMJR  =   - DMJR                                              INSI2010
         PRINT  100, TOTA                                              INSI2020
         NUMTOT  =  NUMTOT - NSUB                                      INSI2030
         PRINT  110, L1, NUMTOT                                        INSI2040
         CALL FILEDT (L1, IRUN, ITOT, CTLPT, NUMTOT)                   INSI2050
 1230    CONTINUE                                                      INSI2060
C                                                                      INSI2070
 1240  CONTINUE                                                        INSI2080
```

```
C                                                                       INSI2090
C                                                                       INSI2100
   10  FORMAT  ( 1H1    )                                              INSI2110
   20  FORMAT  ( 1H1,  //// , 57X *LEACHANT FLOW SIMULATION* //// 15X   INSI2120
     1   *SIMULATION NUMBER * I3 * CONDUCTED FOR * F8.1 * HOURS, AT *   INSI2130
     2   F8.1 * HOURS AFTER START OF PUMPING OPERATION. * // 15X        INSI2140
     3   *AQUIFER IS * 2A8, 35X *WELL RADIUS * F4.1 * INS.* // 15X      INSI2150
     4   *HEAD CHANGE IS TIME * 2A8, 26X *TIME INTERVAL OF PARTICLE FLO*INSI2160
     5   *W EVALUATION * I5 * HRS.* // 15X *AQUIFER POROSITY = * F10.8, INSI2170
     6   33X *AQUIFER THICKNESS = * F7.2 * FT. * // 15X *AQUIFER PERMEA*INSI2180
     7   *BILITY* 42X *DEPTH OF FLOW ANALYSIS (Z) = * F7.2 * FT. * / 25XINSI2190
     8   *(MAJOR AXIS) = * F10.6 * DARCYS* / 25X *(MINOR AXIS) = * F10.6INSI2200
     9   * DARCYS* 20X *BOUNDARIES OF AQUIFER ANALYSIS (FT.)* / 29X     INSI2210
     9   *VERTICAL = * F10.6 * DARCYS* / 77X *X-MAX* 5X,  *X-MIN* 5X    INSI2220
     9   *Y-MAX* 5X *Y-MIN* / 15X *GROUNDWATER VELOCITY = * F10.1       INSI2230
     9   * FT./YR. * 15X4(F8.1, 3X) / 26X *DIRECTION = * F6.2          INSI2240
     9   * DEGREES (FROM POS. X-AXIS)* ////  )                         INSI2250
   30  FORMAT  ( 50X *WELLS OPERATING AT TIME = * F8.1 * HOURS.* /// 28XINSI2260
     1   *LOCATION(FT.)* 7X *PUMPING RATE(GAL/MIN)* 4X *SCREEN LOCATION*INSI2270
     2   *(FT.)* 4X *WELL START* 8X *WELL STOP* / 15X *WELL NO.* 3X     INSI2280
     3   *X-COORD* 3X *Y-COORD* 34X *TOP    BOTTOM* 7X *TIME(HRS.)* 7X  INSI2290
     4   *TIME(HRS.)*  )                                                INSI2300
   40  FORMAT  ( 16XI4, 3X, F8.1, 3X, F8.1, 10X, F7.2, 1X, A8, 6X, F7.2,INSI2310
     1   3X, F7.2, 7X, F8.1, 10X, F8.1 /  )                             INSI2320
   50  FORMAT  (  // 53X *SPECIAL POINTS TO BE EVALUATED* // 50X        INSI2330
     1   *POINT NO.* 14X *LOCATION(FT.)* / 72X *X-COORD* 3X *Y-COORD* //INSI2340
     2   )                                                              INSI2350
   60  FORMAT  ( 50X, I4, 16X, F8.1, 3X, F8.1 /  )                     INSI2360
   70  FORMAT  (  //// 51X *EVALUATION OF SPECIAL INTEREST POINTS* ///  INSI2370
     1   26X,  *X CO-ORD* 10X,  *Y CO-ORD* 10X,  *DRAWDOWN* 11X,        INSI2380
     2   *VELOCITY* 5X,  *DIRECTION OF FLOW* / , 28X,  *(FEET)* 12X,    INSI2390
     3   *(FEET)* 12X,  *(FEET)* 11X,  *(FEET/SEC.)* 7X,  *(DEGREES)* //INSI2400
     4   )                                                              INSI2410
   80  FORMAT  ( 1H0, 20X, 5(E15.5, 3X) )                              INSI2420
   90  FORMAT  ( 5F10.5 )                                              INSI2430
  100  FORMAT  (  // 30X,  *TOTAL TWO DIMENSIONAL AREA SWEPT BY THIS WE*INSI2440
     1   *LL CONFIGURATION = * F15.2 * SQ.FT.*  )                        INSI2450
  110  FORMAT  (  /// 30X, I5,  * TOTAL STREAMLINE COORDINATES OUTPUT T*INSI2460
     1   *O TAPE 6.* / 30X *THE FINAL * I5 * ORDERED PAIRS ARE WELL COO*INSI2470
     2   *RDINATES AND SPECIAL INTEREST POINTS.*  )                      INSI2480
C                                                                       INSI2490
      END                                                               INSI2500
```

```
C////////////////////////////////////////////////////////////////////////INPT  10
      SUBROUTINE  INPT (NUMODD, ODDPT, CTLPT, START)                     INPT  20
C***** INPUT MATRIX SPECIFICATIONS,OUTPUT REQUIREMENTS,AQUIFER           INPT  30
C***** CHARACTERISITCS,WELL PENETRATION DATA,DEPTH OF ANALYSIS.          INPT  40
C***** WELL(1,I) = FLOW RATE (< 0 INDICATES INPUT WELL).                INPT  50
C*****                    (> 0 INDICATES OUTPUT WELL).                 INPT  60
C***** WELL(2,I) = PENETRATION DEPTH.                                   INPT  70
C***** WELL(3,I) = X COORDINATE.                                        INPT  80
C***** WELL(4,I) = YCOORDINATE.                                         INPT  90
C***** WELL(5,I) = WELL PUMPING START TIME.                             INPT 100
C***** WELL(6,I) = WELL PUMPING STOP TIME.                              INPT 110
C***** WELL(7,I) = WELL NUMBER                                          INPT 120
      COMMON  /PIE/  PI                                                INPT 130
      COMMON  /WATER/  WATRX, WATRY, WATDR, WATRV                      INPT 140
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME       INPT 150
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN             INPT 160
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,  INPT 170
     1   N1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM              INPT 180
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ              INPT 190
      COMMON  /LINES/  OUTOPT, MAXID, DISTME, CUTOFF                   INPT 200
      DIMENSION   CTLPT (2,100), ITME (10), ODDPT (9,50), TIME (10),   INPT 210
     1   WELL (9,50)                                                   INPT 220
      INTEGER      AQR, DISTME, OUTOPT                                 INPT 230
      REAL         K, KMJR, KMNR, KPRIME, KX, KY, KZ, K1, N, N1        INPT 240
      READ (5,   10)  ITOT, AQR, NUMODD, NLINES, ITEREND, NUMWL, MAXID,INPT 250
     1   DISTME                                                        INPT 260
      READ (5,   20)  B, Z, RADIUS, GAMMA, TERM                        INPT 270
      READ (5,   30)  ALPHA, BETA                                      INPT 280
      READ (5,   20)  N, KMJR, DMJR, KMNR, ZK                          INPT 290
      IF  (KMJR .EQ. KMNR)  DMJR  =  0.0                               INPT 300
      KX  =  KMJR * .3169E - 4                                         INPT 310
      KY  =  KMNR * .3169E - 4                                         INPT 320
      KZ  =  ZK * .3169E - 4                                           INPT 330
      READ (5,   20)  XMIN, XMAX, YMIN, YMAX                           INPT 340
      READ (5,   20)  WATDR, WATRV                                     INPT 350
      DISTME  =  DISTME * 3600.                                        INPT 360
      IF  (DISTME .LT. 3600.)  DISTME  =  3600.                        INPT 370
      START  =  1.0E12                                                 INPT 380
        DO 1000  J  =  1, NUMWL                                        INPT 390
        READ (5,   20)   (WELL(I, J), I=1, 8)                          INPT 400
        CTLPT (1, J)  =  WELL (3, J)                                   INPT 410
        CTLPT (2, J)  =  WELL (4, J)                                   INPT 420
        WELL (5, J)  =  WELL (5, J) * 3600.                            INPT 430
        WELL (6, J)  =  WELL (6, J) * 3600.                            INPT 440
        WELL (7, J)  =  WELL (7, J) / 7.4805 / 60.0                    INPT 450
        WTIME  =  AMIN1 (WELL(5, J), START)                            INPT 460
1000    START  =  WTIME                                                INPT 470
      L  =  0                                                          INPT 480
        DO 1020  J  =  1, ITOT                                         INPT 490
        READ (5,   40)  TIME (J), ITME (J)                             INPT 500
        TIME (J)  =  TIME (J) * 3600.                                  INPT 510
        TIME (J - L)  =  TIME (J)                                      INPT 520
```

```
      ITME (J - L)  =  ITME (J)                                                 INPT 530
        DO 1010  JJ  =  1, NUMWL                                                INPT 540
        IF (WELL(5,JJ) .LT. TIME(J) .AND. WELL(6,JJ) .GE. TIME(J))              INPT 550
     1    GO TO 1020                                                           INPT 560
 1010   CONTINUE                                                               INPT 570
      L  =  L + 1                                                             INPT 580
      TIME (J)  =  TIME (J) / 3600.                                           INPT 590
      PRINT   50, TIME (J)                                                    INPT 600
 1020 CONTINUE                                                                INPT 610
      ITOT  =  ITOT - L                                                       INPT 620
      IF (ITOT .LE. 0)  STOP                                                  INPT 630
 1030 IF (AQR .GT. 1)  READ (5,   20)  BPRIME, KPRIME                         INPT 640
      KPRIME  =  KPRIME * .3169E - 4                                          INPT 650
 1040 IF (NUMODD .LE. 0)  GO TO 9999                                          INPT 660
      DO 1050  I  =  1, NUMODD                                                INPT 670
      READ (5,   20)   (ODDPT(J, I), J=1, 8)                                  INPT 680
      ODDPT (2, I)  =   - 1                                                   INPT 690
      CTLPT (1, I + NUMWL)  =  ODDPT (3, I)                                   INPT 700
 1050 CTLPT (2, I + NUMWL)  =  ODDPT (4, I)                                   INPT 710
C                                                                             INPT 720
      GO TO 9999                                                              INPT 730
C                                                                             INPT 740
      GO TO 9999                                                              INPT 750
C                                                                             INPT 760
 9999 RETURN                                                                  INPT 770
C                                                                             INPT 780
   10 FORMAT  ( 15I5 )                                                        INPT 790
   20 FORMAT  ( 8F10.0 )                                                      INPT 800
   30 FORMAT  ( 5E15.5 )                                                      INPT 810
   40 FORMAT  ( F10.0, I10 )                                                  INPT 820
   50 FORMAT  ( 1H1,  //////  *NO WELLS ARE PUMPING AT TIME* F10.2             INPT 830
     1   *HOURS--FLOW EVALUATION SCHEDULED AT THIS TIME HAS BEEN SKIPPE*INPT 840
     2   *D*  )                                                               INPT 850
C                                                                             INPT 860
      END                                                                     INPT 870
```

```
C////////////////////////////////////////////////////////////////////////DELT  10
      SUBROUTINE  DELTAH (DELH, FOFT, IRUN)                              DELT  20
C***** DELTAH MAKES A DETERMINATION OF WHICH ROUTINE IS TO BE USED IN     DELT  30
C***** EVALUATING CHANGE IN HEAD.  ALSO DETERMINES THE RELATIVE          DELT  40
C***** IMPORTANCE OF EACH WELL IN THE 5-SPOT PATTERN, IN DETERMINING THEDELT  50
C***** CHANGE IN HEAD AT A GIVEN GRID NODE.                             DELT  60
      COMMON  /PIE/  PI                                                  DELT  70
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME         DELT  80
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN               DELT  90
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,    DELT 100
     1   R1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM                DELT 110
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                DELT 120
      DIMENSION   DELPART (50), ITME (10), R (50), TIME (10), U (50),    DELT 130
     1   WELL (9,50)                                                     DELT 140
      INTEGER     AQR                                                    DELT 150
      REAL        K, KMJR, KMNR, KPRIME, KX, KY, KZ                      DELT 160
        DO 1000  I  =  1, NUMWL                                          DELT 170
        R (I)  =  SQRT ( (X - WELL(3, I)) *  * 2 + (KX / KY) * (Y -      DELT 180
     1     WELL(4, I)) *  * 2)                                           DELT 190
1000    U (I)  =  FOFT / KX * R (I) *  * 2                               DELT 200
      IF (ITME(IRUN) .GT. 1)  GO TO 1020                                 DELT 210
      IF (AQR .GT. 1)  GO TO 1010                                        DELT 220
      CALL SSCONF (DELPART, R, IRUN)                                     DELT 230
      GO TO 1040                                                         DELT 240
1010  CALL SSLEAKY (DELPART, R, IRUN)                                    DELT 250
      GO TO 1040                                                         DELT 260
1020  IF (AQR .GT. 1)  GO TO 1030                                        DELT 270
      CALL TMECONF (DELPART, R, U, IRUN)                                 DELT 280
      GO TO 1040                                                         DELT 290
1030  CALL TMELKY (DELPART, R, U, IRUN)                                  DELT 300
1040  DELH  =  0                                                        DELT 310
        DO 1050  I  =  1, NUMWL                                          DELT 320
1050    DELH  =  DELH + DELPART (I)                                      DELT 330
9999  RETURN                                                             DELT 340
      END                                                                DELT 350
```

```
C////////////////////////////////////////////////////////////////////////VXVY  10
      SUBROUTINE  VXVY (VX, VY, VRAD, THETA, XPART, YPART)               VXVY  20
C***** VELOCITY AND DIRECTION OF FLOW OF PARTICLES IS CALCULATED          VXVY  30
C***** HERE. USED IN DETERMINATION OF STREAMLINES.                        VXVY  40
      COMMON  /PIE/  PI                                                  VXVY  50
      COMMON  /WATER/  WATRX, WATRY, WATDR, WATRV                        VXVY  60
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,    VXVY  70
     1   R1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM                VXVY  80
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                VXVY  90
C     VELOCITY IN X AND Y.                                               VXVY 100
C***** ADD GROUNDWATER FLOW VECTOR TO VELOCITY VECTORS.                   VXVY 110
      DIMENSION   ITME (10)                                              VXVY 120
      REAL        K, KX, KY, KZ, N                                       VXVY 130
      VX  =   - KX * XPART / N + WATRX                                   VXVY 140
      VY  =   - KY * YPART / N + WATRY                                   VXVY 150
      VRAD  =  SQRT (VX *  * 2 + VY *  * 2)                              VXVY 160
      IF  (VX .GE. 0 .AND. VY .GE. 0)  ADD  =  0                         VXVY 170
      IF  (VX .GT. 0 .AND. VY .LT. 0)  ADD  =  0                         VXVY 180
      IF  (VX .LT. 0 .AND. VY .GE. 0)  ADD  =  PI                        VXVY 190
      IF  (VX .LE. 0 .AND. VY .LT. 0)  ADD  =  PI                        VXVY 200
      IF (VX .NE. 0.0)  GO TO 1000                                       VXVY 210
      ETA  =  PI / 2.0 + ADD                                             VXVY 220
      GO TO 1010                                                         VXVY 230
1000  TTAN  =  VY / VX                                                   VXVY 240
      ETA  =  ATAN (TTAN) + ADD                                          VXVY 250
1010  THETA  =  ETA * 180. / PI                                          VXVY 260
1020  IF (THETA .GE. 0.)  GO TO 1030                                     VXVY 270
      THETA  =  360. + THETA                                             VXVY 280
      GO TO 1020                                                         VXVY 290
1030  CONTINUE                                                           VXVY 300
9999  RETURN                                                             VXVY 310
      END                                                                VXVY 320
```

```
C////////////////////////////////////////////////////////////////////////////BESS  10
      FUNCTION  BESSK (BESSIN)                                                BESS  20
C                                                                             BESS  30
C     K BESSEL FUNCTION APPROXIMATION.                                        BESS  40
C                                                                             BESS  50
      IF (BESSIN .GE. 2.)   1010, 1000                                        BESS  60
 1000 BESSK  =   - ALOG (BESSIN / 2.) * BESSI (BESSIN) - .57721566 +          BESS  70
     1   .42278420 * ((BESSIN / 2.) *  * 2) + .23069756 * ( (BESSIN /        BESS  80
     2   2.) *  * 4) + .0348859 * ( (BESSIN / 2.) *  * 6) + .00262698 * BESS  90
     3   ( (BESSIN / 2.) *  * 8) + .0001075 * ( (BESSIN / 2.) *  * 10) +BESS 100
     4   .0000074 * ( (BESSIN / 2.) *  * 12)                                 BESS 110
      GO TO 1020                                                              BESS 120
 1010 BESSK  =   (1. / (SQRT(BESSIN) * EXP(BESSIN))) * (1.25331414 -          BESS 130
     1   .07832358 * (2. / BESSIN) + .02189568 * ( (2. / BESSIN) *  * 2)BESS 140
     2   - .01062446 * ( (2. / BESSIN) *  * 3) + .00587872 * ( (2. /         BESS 150
     3   BESSIN) *  * 4) - .00251540 * ( (2. / BESSIN) *  * 5) +             BESS 160
     4   .00053208 * ( (2. / BESSIN) *  * 6))                                BESS 170
 1020 CONTINUE                                                                BESS 180
 9999 RETURN                                                                  BESS 190
      END                                                                     BESS 200
```

```
C////////////////////////////////////////////////////////////////////////BESS  10
      FUNCTION  BESSI (BESSIN1)                                          BESS  20
C                                                                        BESS  30
C     I BESSEL FUNCTION APPROXIMATION.                                   BESS  40
C                                                                        BESS  50
      BESSIN  =  BESSIN1 / 3.75                                          BESS  60
      IF (BESSIN1 .LT. -3.75 .OR. BESSIN1 .GT. 3.75)  GO TO 1010         BESS  70
 1000 BESSI  =  1. + 3.5156229 * (BESSIN *  * 2) + 3.0899424 * (BESSIN BESS  80
     1   *  * 4) + 1.2067492 * (BESSIN *  * 6) + .2659732 * (BESSIN *  *BESS  90
     2   8) + .0360768 * (BESSIN *  * 10) + .0045813 * (BESSIN *  * 12) BESS 100
      GO TO 1020                                                         BESS 110
 1010 BESSI  =  (EXP(BESSIN1) / SQRT(BESSIN1)) * (.39894228 +            BESS 120
     1   .01328592 * (BESSIN *  * ( - 1)) + .00225319 * (BESSIN *  * ( -BESS 130
     2   2)) - .00157565 * (BESSIN *  * ( - 3)) + .00916281 * (BESSIN * BESS 140
     3   * ( - 4)) - .02057706 * (BESSIN *  * ( - 5)) + .02635537 *     BESS 150
     4   (BESSIN *  * ( - 6)) - .01647633 * (BESSIN *  * ( - 7)) +      BESS 160
     5   .00392377 * (BESSIN *  * ( - 8)))                              BESS 170
 1020 CONTINUE                                                           BESS 180
 9999 RETURN                                                             BESS 190
      END                                                                BESS 200
```

```
C////////////////////////////////////////////////////////////////////////////ODDP  10
      SUBROUTINE  ODDPLOT (TMEFLO, NUMODD, ODDLIST, ODDPT, IRUN)              ODDP  20
      COMMON  /PIE/  PI                                                       ODDP  30
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,         ODDP  40
     1   R1, SS, ITOT, AQR, BPRIME, KPRIME, ITEPEND, TERM                    ODDP  50
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME              ODDP  60
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN                    ODDP  70
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                     ODDP  80
      COMMON  /WATER/  WATRX, WATRY, WATDR, WATRV                             ODDP  90
C***** ALL ODDPOINTS (NOT ON THE ESTABLISHED MATRIX) ARE EVALUATED HERE.ODDP 100
C***** INCLUDES POINTS REQUIRED FOR STREAMLINE EVALUATION AS WELL AS     ODDP 110
C*****  OTHER SPECIAL INTEREST POINTS.                                   ODDP 120
      DIMENSION   ITME (10), ODDLIST (9,50), ODDPT (9,50), TIME (10),     ODDP 130
     1   WELL (9,50), XLIST (2), YLIST (2)                                ODDP 140
      REAL        N                                                       ODDP 150
      FOFT  =  GAMMA * (ALPHA + N * BETA) / (4.0 * TMEFLO)                ODDP 160
        DO 1020  JJ  =  1, NUMODD                                         ODDP 170
        IY  =  1                                                          ODDP 180
        Y  =  ODDPT (4, JJ)                                               ODDP 190
          DO 1000  IX  =  1, 2                                            ODDP 200
          X  =  ODDPT (3, JJ) + 2. / 3. * IX * 10. / SIZE - 10. / SIZE    ODDP 210
1000      CALL DELTAH (XLIST(IX), FOFT, IRUN)                             ODDP 220
        XPART  =   (XLIST(1) - XLIST(2)) / (2. / 3. * 10. / SIZE)         ODDP 230
        IX  =  1                                                          ODDP 240
        X  =  ODDPT (3, JJ)                                               ODDP 250
          DO 1010  IY  =  1, 2                                            ODDP 260
          Y  =  ODDPT (4, JJ) + 2. / 3. * IY * 10. / SIZE - 10. / SIZE    ODDP 270
1010      CALL DELTAH (YLIST(IY), FOFT, IRUN)                             ODDP 280
        YPART  =   (YLIST(1) - YLIST(2)) / (2. / 3. * 10. / SIZE)         ODDP 290
        CALL VXVY (VX, VY, VRAD, THETA, XPART, YPART)                     ODDP 300
        Y  =  ODDPT (4, JJ)                                               ODDP 310
        CALL DELTAH (DELH, FOFT, IRUN)                                    ODDP 320
        ODDLIST (1, JJ)  =  XPART                                         ODDP 330
        ODDLIST (2, JJ)  =  YPART                                         ODDP 340
        ODDLIST (3, JJ)  =  ODDPT (3, JJ)                                 ODDP 350
        ODDLIST (4, JJ)  =  ODDPT (4, JJ)                                 ODDP 360
        ODDLIST (5, JJ)  =  DELH                                          ODDP 370
        ODDLIST (6, JJ)  =  VX                                            ODDP 380
        ODDLIST (7, JJ)  =  VY                                            ODDP 390
        ODDLIST (8, JJ)  =  VRAD                                          ODDP 400
1020    ODDLIST (9, JJ)  =  THETA                                         ODDP 410
9999  RETURN                                                              ODDP 420
      END                                                                 ODDP 430
```

```
C////////////////////////////////////////////////////////////////////////////STRM  10
      SUBROUTINE  STRMLNE (N, L1, STRMST, NUMPT, RADIUS, NLINES, ANGL1,STRM  20
     1   TIME, IRUN, WELL, NUMWL, LIST, Z)                                 STRM  30
C                                                                          STRM  40
C     THIS SUBROUTINE CALCULATES THE VALUE OF STREAMLINES                  STRM  50
C     (DIRECTION OF FLOW) OF PARTICLES AT A SUFFICIENT NUMBER OF POINTS    STRM  60
C     TO ENABLE CONSTRUCTION OF A CONTOUR MAP.                             STRM  70
C***** THE MINIMUM TIME INTERVAL BETWEENEVALUATION POINTS IS 1 HOUR.       STRM  80
      COMMON  /PIE/  PI                                                    STRM  90
      COMMON  /CROSS/  NCSPT, CSTRM (3000, 5), LFLAG                       STRM 100
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN                 STRM 110
      COMMON  /LINES/  OUTOPT, MAXID, DISTME, CUTOFF                       STRM 120
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                  STRM 130
      COMMON  /WATER/  WATRX, WATRY, WATDR, WATRV                          STRM 140
      DIMENSION   ANGL1 (50), COEF (6), COORD (3), GAL (50), POINT1        STRM 150
     1   (9), POINT2 (9), POINT3 (9), STREAM (900,10), STRMLST (9),        STRM 160
     2   STRMPT1 (9), STRMPT2 (9), STRMST (9,50), TIME (10), VELDFX (3),STRM 170
     3   VELDFY (3), WELL (9,50)                                           STRM 180
      INTEGER     DISTME, OUTOPT, V, W                                     STRM 190
      REAL        KX, KY, KZ, MAX, N                                       STRM 200
      DATA    COORD / 8HX CO-ORD, 8HY CO-ORD, 8HZ CO-ORD  /                STRM 210
      LNEND  =  0                                                          STRM 220
      NN  =  0                                                             STRM 230
      MAX  =  0.0                                                          STRM 240
      HRS  =  TIME (IRUN) / 3600.                                          STRM 250
      PRINT   10, HRS                                                      STRM 260
      TINTT  =  TINT  =  TINT2  =  TINT3  =  500.                          STRM 270
      WELMX  =  0.0                                                        STRM 280
C****  FIND THE WELL WITH GREATEST Q.                                      STRM 290
        DO 1000  I  =  1, NUMWL                                            STRM 300
        IF (WELL(5,I) .GE. TIME(IRUN) .OR. WELL(6,I) .LT. TIME(IRUN))      STRM 310
     1     GO TO 1000                                                      STRM 320
        IF (ABS(WELL(7,I)) .LT. WELMX)  GO TO 1000                         STRM 330
        WELMX  =  WELL (7, I)                                              STRM 340
        MX  =  I                                                           STRM 350
 1000   CONTINUE                                                           STRM 360
C**** DETERMINE SUITABLY SMALL INITIAL STREAMLINE INTERVAL.                STRM 370
      MFLAG  =  1                                                          STRM 380
      IF  (WELL(7, MX) .GT. 0.0)  MFLAG  =  - 1                            STRM 390
      WELL (7, MX)  =  MFLAG * WELL (7, MX)                                STRM 400
      POINT1 (3)  =  WELL (3, MX) + RADIUS + 100. / SIZE                   STRM 410
      POINT1 (4)  =  WELL (4, MX)                                          STRM 420
      CALL ODDPLOT (TINTT, 1, STRMLST, POINT1, IRUN)                       STRM 430
      VELDFX (2)  =  STRMLST (6)                                           STRM 440
      VELDFY (2)  =  STRMLST (7)                                           STRM 450
      TINTR  =  TINT                                                       STRM 460
 1010 POINT2 (3)  =  POINT1 (3) + VELDFX (2) * TINT                        STRM 470
      POINT2 (4)  =  POINT1 (4) + VELDFY (2) * TINT                        STRM 480
      CALL ODDPLOT (TINTT, 1, STRMLST, POINT2, IRUN)                       STRM 490
      V2  =  SQRT (VELDFX(2) *  * 2 + VELDFY(2) *  * 2)                    STRM 500
      V1  =  SQRT (STRMLST(6) *  * 2 + STRMLST(7) *  * 2)                  STRM 510
      VRA  =  V1 / V2                                                      STRM 520
```

```
      IF  (V1 .GT. V2)  VRA  =  V2 / V1                                  STRM 530
      IF (VRA .LT. .95)  GO TO 1020                                      STRM 540
      IF (VRA .GT. .99)  GO TO 1030                                      STRM 550
      GO TO 1040                                                         STRM 560
 1020 IF (TINT .NE. TINTB)  GO TO 1040                                   STRM 570
      TINT  =  TINT - TINT / 2.0                                         STRM 580
      TINTB  =  TINT                                                     STRM 590
      GO TO 1010                                                         STRM 600
 1030 TINT  =  TINT + TINT / 2.0                                         STRM 610
      GO TO 1010                                                         STRM 620
 1040 TMIN  =  TINT / 4.0                                                STRM 630
      TINTS  =  TINT                                                     STRM 640
      WELL (7, MX)  =  MFLAG * WELL (7, MX)                              STRM 650
C**** FIND SERIES REMAINDER FOR LATER COMPARISON TO OTHER TIME INTERVALSSTRM 660
      POINT3 (3)  =  POINT2 (3) + STRMLST (6) * TINT                     STRM 670
      POINT3 (4)  =  POINT2 (4) + STRMLST (7) * TINT                     STRM 680
      VELDFX (3)  =  VELDFX (2)                                          STRM 690
      VELDFY (3)  =  VELDFY (2)                                          STRM 700
      VELDFX (2)  =  STRMLST (6)                                         STRM 710
      VELDFY (2)  =  STRMLST (7)                                         STRM 720
      CALL ODDPLOT (TINTT, 1, STRMLST, POINT3, IRUN)                     STRM 730
      VL3  =  SQRT (VELDFX(3) *  * 2 + VELDFY(3) *  * 2)                 STRM 740
      VL2  =  SQRT (VELDFX(2) *  * 2 + VELDFY(2) *  * 2)                 STRM 750
      VL1  =  STRMLST (8)                                                STRM 760
      ACC1  =   (VL1 - VL2) / TINT                                       STRM 770
      ACC2  =   (VL2 - VL3) / TINT                                       STRM 780
      APRIME  =  ABS (ACC1 - ACC2) / TINT                                STRM 790
      RNORM  =  APRIME * TINT *  * 3 / 6.0                               STRM 800
        DO 1360  W  =  1, NUMPT                                          STRM 810
        IWELL  =  STRMST (8, W)                                          STRM 820
        WELL (9, W)  =  0.0                                              STRM 830
        ST3W  =  STRMST (3, W)                                           STRM 840
        ST4W  =  STRMST (4, W)                                           STRM 850
        CALL CNVRT (ST3W, ST4W, DMJR)                                    STRM 860
        XCHK  =  ST3W * SIZE / 1000. + XMIN                              STRM 870
        YCHK  =  ST4W * SIZE / 1000. + YMIN                              STRM 880
        IF (XCHK .GE. XMAX .OR. XCHK .LE. XMIN)  GO TO 1360              STRM 890
        IF (YCHK .GE. YMAX .OR. YCHK .LE. YMIN)  GO TO 1360              STRM 900
        TOTGL  =  - TIME (IRUN) * STRMST (7, W) * 7.4805 * (SIZE /       STRM 910
     1    1000.) *  * 3                                                 STRM 920
        IF (STRMST(5,W) .GE. TIME(IRUN) .OR. STRMST(6,W) .LT.            STRM 930
     1    TIME(IRUN))  GO TO 1360                                        STRM 940
        IF (STRMST(7,W) .GE. 0)  GO TO 1360                              STRM 950
          DO 1050  I  =  1, NUMPT                                        STRM 960
 1050     GAL (I)  =  0.0                                                STRM 970
        TPCNT  =  0.0                                                    STRM 980
        ESC  =  100.                                                     STRM 990
        WTOT  =  0.0                                                     STRM1000
          DO 1340  I  =  1, NLINES                                       STRM1010
          AREA  =  0.0                                                   STRM1020
          LIST  =  LIST + 1                                              STRM1030
          JFLAG  =  0                                                    STRM1040
```

```
      SUMT  =  TINTT                                                    STRM1050
      TINT  =  TINT2  =  TINTS                                          STRM1060
      ACX  =  ACY  =  0.0                                               STRM1070
      IVEC  =  0                                                        STRM1080
      J  =  1                                                           STRM1090
      V  =  1                                                           STRM1100
      SUMINT  =  0.0                                                    STRM1110
      ANGLE  =  360.0 * (I - 1) / NLINES + ANGL1 (W)                    STRM1120
      RADIAN  =  ANGLE * PI / 180.                                      STRM1130
      STREAM (J, 1)  =  STRMPT1 (3)  =  STRMST (3, W) + (RADIUS +       STRM1140
     1    1000. / SIZE) * COS (RADIAN)                                   STRM1150
      STREAM (J, 2)  =  STRMPT1 (4)  =  STRMST (4, W) + (RADIUS +       STRM1160
     1    1000. / SIZE) * SIN (RADIAN)                                   STRM1170
      STREAM (J, 7)  =  LIST                                            STRM1180
      STREAM (J, 3)  =  Z / SQRT (KX / KZ) * SIZE / 1000.               STRM1190
      CSTRM (V, 1)  =  STREAM (J, 1)                                    STRM1200
      CSTRM (V, 2)  =  STREAM (J, 2)                                    STRM1210
      CSTRM (V, 4)  =  1.0                                              STRM1220
      CALL CNVRT (STREAM(J, 1), STREAM(J, 2), DMJR)                     STRM1230
      STREAM (J, 1)  =  STREAM (J, 1) * SIZE / 1000. + XMIN             STRM1240
      STREAM (J, 2)  =  STREAM (J, 2) * SIZE / 1000. + YMIN             STRM1250
      CALL ODDPLOT (TINTT, 1, STRMLST, STRMPT1, IRUN)                   STRM1260
      STREAM (J, 4)  =  STRMLST (8) * SIZE / 1000.                      STRM1270
      STREAM (J, 5)  =  STRMLST (5) * SIZE / 1000.                      STRM1280
      IJK  =  0                                                         STRM1290
      COEF (1)  =  STREAM (J, 1)                                        STRM1300
      COEF (2)  =  STREAM (J, 2)                                        STRM1310
      COEF (3)  =  STRMLST (6) * SIZE / 1000.                           STRM1320
      COEF (4)  =  STRMLST (7) * SIZE / 1000.                           STRM1330
      COEF (5)  =  STREAM (J, 4)                                        STRM1340
      COEF (6)  =  STREAM (J, 5)                                        STRM1350
      WRITE (7,   20)   (COEF(LMN), LMN=1, 6), IJK, I, IWELL, NN        STRM1360
      DIAM1  =  0.0                                                     STRM1370
      J  =  J + 1                                                       STRM1380
C***** CALCULATE ENOUGH STREAMLINE POINTS TO ENABLE APPROX. OF          STRM1390
C***** VELOCITY,ACCEL., AND ACCEL(PRIME).                              STRM1400
      DO 1060  JFLAG  =  1, 3                                           STRM1410
      SUMT  =  SUMT + TINT                                              STRM1420
      CALL ODDPLOT (TINTT, 1, STRMLST, STRMPT1, IRUN)                   STRM1430
      ACX  =  (STRMLST(6) - VELDFX(2)) / TINT                           STRM1440
      ACY  =  (STRMLST(7) - VELDFY(2)) / TINT                           STRM1450
      STRMPT2 (3)  =  STRMPT1 (3) + STRMLST (6) * TINT + ACX /          STRM1460
     1    2.0 * TINT *  * 2                                            STRM1470
      STRMPT2 (4)  =  STRMPT1 (4) + STRMLST (7) * TINT + ACY /          STRM1480
     1    2.0 * TINT *  * 2                                            STRM1490
      VELDFX (3)  =  VELDFX (2)                                         STRM1500
      VELDFY (3)  =  VELDFY (2)                                         STRM1510
      VELDFX (2)  =  STRMLST (6)                                        STRM1520
      VELDFY (2)  =  STRMLST (7)                                        STRM1530
      VL2  =  SQRT (VELDFX(2) *  * 2 + VELDFY(2) *  * 2)                STRM1540
      VL3  =  SQRT (VELDFX(3) *  * 2 + VELDFY(3) *  * 2)                STRM1550
      ACC2  =  (VL2 - VL3) / TINT                                       STRM1560
```

```
                XTINT  =  TINT                                                  STRM1570
                IF (JFLAG .EQ. 3)  GO TO 1060                                   STRM1580
                STREAM (J, 1)  =  STRMPT1 (3)  =  STRMPT1 (3) + STRMLST (6)STRM1590
     1            * TINT + ACX / 2.0 * TINT *  * 2                             STRM1600
                STREAM (J, 2)  =  STRMPT1 (4)  =  STRMPT1 (4) + STRMLST (7)STRM1610
     1            * TINT + ACY / 2.0 * TINT *  * 2                             STRM1620
 1060           CONTINUE                                                        STRM1630
              SUMINT  =  SUMT                                                   STRM1640
              CSTRM (V, 5)  =  SUMT                                             STRM1650
              V  =  V + 1                                                       STRM1660
              CSTRM (V, 1)  =  STRMPT2 (3)                                      STRM1670
              CSTRM (V, 2)  =  STRMPT2 (4)                                      STRM1680
              JFLAG  =  1                                                       STRM1690
              TINT2  =  TINT                                                    STRM1700
 1070         CONTINUE                                                        STRM1710
              TINTA  =  TINTB  =  TINT                                          STRM1720
 1080         CONTINUE                                                        STRM1730
C***** BEGIN CALCULATION OF STREAMLINE POINTS WITH TIME INTERVAL                STRM1740
C***** DEPENDENT UPON ACCEL(PRIME).                                            STRM1750
              CALL ODDPLOT (SUMINT, 1, STRMLST, STRMPT2, IRUN)                  STRM1760
              ACX1  =   (STRMLST(6) - VELDFX(2)) / TINT                         STRM1770
              ACY1  =   (STRMLST(7) - VELDFY(2)) / TINT                         STRM1780
              ACX2  =   (VELDFX(2) - VELDFX(3)) / TINT2                         STRM1790
              ACY2  =   (VELDFY(2) - VELDFY(3)) / TINT2                         STRM1800
              AX  =   (ACX1 * (2. * TINT + TINT2) - ACX2 * TINT) / (TINT + STRM1810
     1          TINT2)                                                         STRM1820
              AY  =   (ACY1 * (2. * TINT + TINT2) - ACY2 * TINT) / (TINT + STRM1830
     1          TINT2)                                                         STRM1840
              VL1  =  SQRT (STRMLST(6) *  * 2 + STRMLST(7) *  * 2)            STRM1850
              ACC1  =   (VL1 - VL2) / TINT                                      STRM1860
C***** CALCULATE REMAINDER AFTER 3 TERM TAYLOR EXPANSION AROUND                 STRM1870
C***** STREAMLINE DISTANCE INTERVALS.                                          STRM1880
              APRIME  =  ABS (ACC1 - ACC2) / (TINT + TINT2) / 2.0               STRM1890
              REMAIN  =  APRIME * TINT *  * 3 / 6.0                             STRM1900
              JFLAG  =  JFLAG + 1                                               STRM1910
C***** NORMALIZE REMAINDER.                                                    STRM1920
              RATIO  =  REMAIN / RNORM                                          STRM1930
              IF (RATIO .LT. 6.0)  GO TO 1090                                   STRM1940
              IF (TINTA .NE. TINT)  GO TO 1110                                  STRM1950
              IF (TINT .LE. TMIN)  GO TO 1110                                   STRM1960
C***** CHECK  TIME INTERVAL TOO LARGE.                                          STRM1970
              TINT  =  TINT - TINT / 5.                                         STRM1980
              TINTA  =  TINT                                                    STRM1990
              GO TO 1100                                                        STRM2000
 1090         IF (RATIO .GE. 3.0)  GO TO 1110                                   STRM2010
              IF (TINTB .NE. TINT)  GO TO 1110                                  STRM2020
C***** CHECK  TIME INTERVAL TOO SMALL.                                          STRM2030
              TINT  =  TINT + TINT / 5.                                         STRM2040
              TINTB  =  TINT                                                    STRM2050
 1100         STRMPT2 (3)  =  STRMPT1 (3) + VELDFX (2) * TINT + AX / 2.0 * STRM2060
     1          TINT *  * 2                                                    STRM2070
              STRMPT2 (4)  =  STRMPT1 (4) + VELDFY (2) * TINT + AY / 2.0 * STRM2080
```

```
     1       TINT *  * 2                                                    STRM2090
             GO TO 1080                                                     STRM2100
C***** SELECT STREAMLINE POINTS AT FIXED INTERVALS FOR PRINTING.            STRM2110
 1110        IF (SUMT+TINT .LT. DISTME)  GO TO 1170                         STRM2120
             XTINT  =  DISTME - SUMT                                        STRM2130
             ST1  =  STREAM (J, 1)  =  STRMPT1 (3) + VELDFX (2) * XTINT +   STRM2140
     1       AX / 2.0 * XTINT *  * 2                                        STRM2150
             ST2  =  STREAM (J, 2)  =  STRMPT1 (4) + VELDFY (2) * XTINT +   STRM2160
     1       AY / 2.0 * XTINT *  * 2                                        STRM2170
             STREAM (J, 3)  =  Z / SQRT (KX / KZ) * SIZE / 1000.           STRM2180
             SUMT  =  TINT - XTINT                                          STRM2190
 1120        CONTINUE                                                       STRM2200
             CSTRM (V, 5)  =  XTINT                                         STRM2210
             V  =  V + 1                                                    STRM2220
             CSTRM (V, 1)  =  STREAM (J, 1)                                 STRM2230
             CSTRM (V, 2)  =  STREAM (J, 2)                                 STRM2240
             CSTRM (V, 4)  =  1.0                                           STRM2250
             IF (V .GT. 3000)  GO TO 1370                                   STRM2260
             CALL CNVRT (STREAM(J, 1), STREAM(J, 2), DMJR)                  STRM2270
             STREAM (J, 1)  =  STREAM (J, 1) * SIZE / 1000. + XMIN          STRM2280
             STREAM (J, 2)  =  STREAM (J, 2) * SIZE / 1000. + YMIN          STRM2290
             STREAM (J, 7)  =  LIST                                         STRM2300
             STREAM (J, 4)  =  VL2 * SIZE / 1000.                           STRM2310
             STREAM (J, 5)  =  STRMLST (5) * SIZE / 1000.                   STRM2320
             IJK  =  1                                                      STRM2330
             COEF (1)  =  STREAM (J, 1)                                     STRM2340
             COEF (2)  =  STREAM (J, 2)                                     STRM2350
             COEF (3)  =  VELDFX (2) * SIZE / 1000.                         STRM2360
             COEF (4)  =  VELDFY (2) * SIZE / 1000.                         STRM2370
             COEF (5)  =  STREAM (J, 4)                                     STRM2380
             COEF (6)  =  STREAM (J, 5)                                     STRM2390
             WRITE (7,   20)   (COEF(LMN), LMN=1, 6), IJK, I, IWELL, NN     STRM2400
C     CALCULATE INCREMENTAL STREAMLINE AREA.                                STRM2410
             WELL1  =  STRMST (7, W) * (SIZE / 1000.) *  * 3                STRM2420
             WELL2  =  STRMST (2, W) * SIZE / 1000.                         STRM2430
             DIAM2  =  ABS (WELL1 / (NLINES * STREAM(J, 4) * WELL2 * N))    STRM2440
             WMAX  =  AMAX1 (DIAM1, DIAM2)                                  STRM2450
             WDIF  =  ABS (DIAM2 - DIAM1)                                   STRM2460
             DST  =  SQRT ((STREAM(J, 1) - STREAM(J - 1, 1)) *  * 2 +       STRM2470
     1       (STREAM(J, 2) - STREAM(J - 1, 2)) *  * 2)                      STRM2480
             STREAM (J, 6)  =  DST * (WMAX - WDIF) + DST * WDIF / 2.0       STRM2490
             DIAM1  =  DIAM2                                                STRM2500
             AREA  =  AREA + STREAM (J, 6)                                  STRM2510
C      STREAMLINE OUTSIDE BOUNDRIES OF ANALYSIS.                            STRM2520
             IF (STREAM(J,1) .GT. XMIN .AND. STREAM(J,1) .LE. XMAX)  GO TOSTRM2530
     1       1130                                                           STRM2540
             IFLAG  =  1                                                    STRM2550
             GO TO 1230                                                     STRM2560
 1130        IF (STREAM(J,2) .GT. YMIN .AND. STREAM(J,2) .LE. YMAX)  GO TOSTRM2570
     1       1140                                                           STRM2580
             IFLAG  =  1                                                    STRM2590
             GO TO 1230                                                     STRM2600
```

```
C***** STREAMLINE TERMINATES WHEN EVALUATION TIME IS REACHED.                    STRM2610
 1140      IF  (IRUN .GT. 1)  TMEOUT  =  TIME (IRUN) - TIME (IRUN - 1)          STRM2620
           IF  (IRUN .EQ. 1)  TMEOUT  =  TIME (IRUN) - STRMST (5, W)            STRM2630
           IF (SUMINT .LE. TMEOUT)  GO TO 1150                                  STRM2640
           IFLAG  =  2                                                          STRM2650
           GO TO 1230                                                           STRM2660
C***** MAXIMUM NUMBER OF STREAMLINE POINTS CALCULATED.                          STRM2670
 1150      IF (J .GT. MAXID)  GO TO 1160                                        STRM2680
           J  =  J + 1                                                          STRM2690
           IF (SUMT .LT. DISTME)  GO TO 1180                                    STRM2700
           XTINT  =  DISTME                                                     STRM2710
           SUMT  =  SUMT - XTINT                                                STRM2720
           ST1  =  STREAM (J, 1)  =  ST1 + VELDFX (2) * XTINT + AX / 2.0STRM2730
     1       * XTINT *  * 2                                                     STRM2740
           ST2  =  STREAM (J, 2)  =  ST2 + VELDFY (2) * XTINT + AY / 2.0STRM2750
     1       * XTINT *  * 2                                                     STRM2760
           STREAM (J, 3)  =  Z / SQRT (KX / KZ) * SIZE / 1000.                  STRM2770
           GO TO 1120                                                           STRM2780
 1160      PRINT   30, MAXID, I                                                 STRM2790
           IFLAG  =  2                                                          STRM2800
           GO TO 1230                                                           STRM2810
 1170      SUMT  =  SUMT + TINT                                                 STRM2820
           CSTRM (V, 5)  =  TINT                                                STRM2830
           GO TO 1190                                                           STRM2840
 1180      CSTRM (V, 5)  =  SUMT                                                STRM2850
 1190      STRMPT1 (3)  =  STRMPT1 (3) + VELDFX (2) * TINT + AX / 2.0 * STRM2860
     1       TINT *  * 2                                                        STRM2870
           STRMPT1 (4)  =  STRMPT1 (4) + VELDFY (2) * TINT + AY / 2.0 * STRM2880
     1       TINT *  * 2                                                        STRM2890
           V  =  V + 1                                                          STRM2900
           CSTRM (V, 1)  =  STRMPT1 (3)                                         STRM2910
           CSTRM (V, 2)  =  STRMPT1 (4)                                         STRM2920
           XV  =  CSTRM (V, 1)                                                  STRM2930
           YV  =  CSTRM (V, 2)                                                  STRM2940
           CALL CNVRT (XV, YV, DMJR)                                            STRM2950
           XV  =  XV * SIZE / 1000. + XMIN                                      STRM2960
           YV  =  YV * SIZE / 1000. + YMIN                                      STRM2970
           IJK  =  0                                                            STRM2980
           COEF (1)  =  XV                                                      STRM2990
           COEF (2)  =  YV                                                      STRM3000
           COEF (3)  =  VELDFX (2) * SIZE / 1000.                               STRM3010
           COEF (4)  =  VELDFY (2) * SIZE / 1000.                               STRM3020
           COEF (5)  =  VL2 * SIZE / 1000.                                      STRM3030
           COEF (6)  =  STRMLST (5) * SIZE / 1000.                              STRM3040
           WRITE (7,   20)   (COEF(LMN), LMN=1, 6), IJK, I, IWELL, NN           STRM3050
           CSTRM (V, 4)  =  0.0                                                 STRM3060
           IF (V .GT. 3000)  GO TO 1370                                         STRM3070
           STRMPT2 (3)  =  STRMPT1 (3) + STRMLST (6) * TINT + AX / 2.0 *STRM3080
     1       TINT *  * 2                                                        STRM3090
           STRMPT2 (4)  =  STRMPT1 (4) + STRMLST (7) * TINT + AY / 2.0 *STRM3100
     1       TINT *  * 2                                                        STRM3110
           VELDFX (3)  =  VELDFX (2)                                            STRM3120
```

```
      VELDFY (3)  =  VELDFY (2)                                          STRM3130
      VELDFX (2)  =  STRMLST (6)                                         STRM3140
      VELDFY (2)  =  STRMLST (7)                                         STRM3150
      VL2  =  SQRT (VELDFX(2) *  * 2 + VELDFY(2) *  * 2)                 STRM3160
      VL3  =  SQRT (VELDFX(3) *  * 2 + VELDFY(3) *  * 2)                 STRM3170
      ACC2  =  (VL2 - VL3) / TINT                                        STRM3180
        DO 1200  KK  =  1, NUMWL                                         STRM3190
        IW7  =  WELL (8, KK)                                             STRM3200
        IF (WELL(7,KK) .LE. 0)  GO TO 1200                               STRM3210
        IF (WELL(5,KK) .GE. TIME(IRUN) .OR. WELL(6,KK) .LT.              STRM3220
     1    TIME(IRUN))  GO TO 1200                                        STRM3230
C***** CHECK   STREAMLINE APPROACH TO OUTPUT WELLS.                      STRM3240
        FARWELL  =  SQRT ((STRMPT2(3) - WELL(3, KK)) *  * 2 +            STRM3250
     1    (STRMPT2(4) - WELL(4, KK)) *  * 2)                             STRM3260
        IF (FARWELL .LE. 2.*RADIUS)  GO TO 1210                          STRM3270
 1200   CONTINUE                                                         STRM3280
      SUMINT  =  SUMINT + TINT                                           STRM3290
      TINT2  =  TINT                                                     STRM3300
      GO TO 1070                                                         STRM3310
 1210 IF (STRMST(2,W) .LT. 0.0)  GO TO 1220                              STRM3320
      WELL (9, KK)  =  WELL (9, KK) + 1.0                                STRM3330
      PTOT  =  TOTGL / NLINES + WELL (7, W) / NLINES * 7.4805 *          STRM3340
     1  SUMINT * (SIZE / 1000.) *  * 3                                  STRM3350
      IF  (TIME(IRUN) .LT. SUMINT)  PTOT  =  0.0                         STRM3360
      GAL (KK)  =  GAL (KK) + PTOT                                       STRM3370
      WTOT  =  WTOT + PTOT                                               STRM3380
      TPCNT  =  WTOT / TOTGL * 100.0                                     STRM3390
      ESC  =  100. - TPCNT                                               STRM3400
 1220 SUMINT  =  SUMINT + FARWELL / STRMLST (8)                          STRM3410
      WELL3  =  WELL (3, KK)                                             STRM3420
      WELL4  =  WELL (4, KK)                                             STRM3430
      CALL CNVRT (WELL3, WELL4, DMJR)                                    STRM3440
      STREAM (J, 3)  =  Z / SQRT (KX / KZ) * SIZE / 1000.                STRM3450
      STREAM (J, 8)  =  FARWELL * SIZE / 1000.                           STRM3460
      STREAM (J, 9)  =  WELL (8, W)                                      STRM3470
      STREAM (J, 10)  =  I                                               STRM3480
      STREAM (J, 1)  =  WELL3 * SIZE / 1000. + XMIN                      STRM3490
      STREAM (J, 2)  =  WELL4 * SIZE / 1000. + YMIN                      STRM3500
      STREAM (J, 7)  =  LIST                                             STRM3510
      STREAM (J, 4)  =  VL2 * SIZE / 1000.                               STRM3520
      STREAM (J, 5)  =  99999.                                           STRM3530
      DST  =  SQRT ((STREAM(J, 1) - STREAM(J - 1, 1)) *  * 2 +           STRM3540
     1  (STREAM(J, 2) - STREAM(J - 1, 2)) *  * 2)                       STRM3550
      STREAM (J, 6)  =  DST * DIAM1 / 2.0                                STRM3560
      AREA  =  AREA + STREAM (J, 6)                                      STRM3570
      IFLAG  =  3                                                        STRM3580
 1230 SUMTME  =  SUMINT / 3600.                                          STRM3590
      STRMST (9, W)  =  STRMST (9, W) + AREA                             STRM3600
      LV  =  V - 1                                                       STRM3610
      WRITE (7,  40)  LV                                                 STRM3620
C*************************************************************************STRM3630
C***** ASSEMBLE STREAMLINE COORDINATES FOR SPECIFIED OUTPUT MEDIUM.      STRM3640
```

```
          ANGLE  =  ANGLE - DMJR * 180. / PI                                  STRM3650
          IF  (ANGLE .GE. 360.)  ANGLE  =  ANGLE - 360.                      STRM3660
          IF (LFLAG .EQ. 0)  GO TO 1240                                      STRM3670
          CALL VFLOW (CSTRM, STREAM, IRUN, SIZE, KX, KZ, V, W, I, KK,        STRM3680
     1      IFLAG)                                                           STRM3690
 1240     PRINT   50, I, IWELL, ANGLE, J                                     STRM3700
          L1  =  L1 + J                                                      STRM3710
          GO TO  ( 1250, 1260, 1270)  IFLAG                                  STRM3720
 1250     PRINT   60, SUMTME                                                 STRM3730
          GO TO 1280                                                         STRM3740
 1260     PRINT   70, SUMTME                                                 STRM3750
          GO TO 1280                                                         STRM3760
 1270     PRINT   80, IW7, SUMTME                                            STRM3770
 1280     ISTOP  =  0                                                        STRM3780
          LOOP  =  J / 13 + 1                                                STRM3790
            DO 1320  INUM  =  1, LOOP                                        STRM3800
            ISTART  =  13 * INUM - 12                                        STRM3810
            IF (LOOP .EQ. INUM)  GO TO 1290                                  STRM3820
            ISTOP  =  ISTART + 12                                            STRM3830
            GO TO 1300                                                       STRM3840
 1290       ISTOP  =  ISTOP + MOD (J, 13)                                    STRM3850
            IF (MOD(J,13) .EQ. 0)  GO TO 1320                                STRM3860
 1300         DO 1310  M  =  1, 3                                            STRM3870
 1310         PRINT   90, COORD (M), (STREAM (L, M), L=ISTART, ISTOP)        STRM3880
 1320       PRINT  100                                                       STRM3890
          PRINT  110, I, AREA                                                STRM3900
 1330     WRITE (6,  120)    ((STREAM(L, M), M=1, 10), L=1, J)               STRM3910
          WRITE (13,  130)    ((STREAM(L, M), M=1, 5), L=1, J)               STRM3920
 1340     CONTINUE                                                           STRM3930
        IF (STRMST(2,W) .LT. 0.0)  GO TO 1360                                STRM3940
        RTIME  =  TIME (IRUN) / 3600.                                        STRM3950
        PRINT  140, IWELL, RTIME                                             STRM3960
        PRINT  150, IWELL                                                    STRM3970
          DO 1350  L  =  1, NUMPT                                            STRM3980
          IF (WELL(7,L) .LE. 0.0)  GO TO 1350                                STRM3990
          JWELL  =  STRMST (8, L)                                            STRM4000
          PRINT  160, JWELL, GAL (L)                                         STRM4010
 1350     CONTINUE                                                           STRM4020
        PRINT  170, IWELL, TPCNT, IWELL, ESC                                 STRM4030
        PRINT  180, IWELL, STRMST (9, W)                                     STRM4040
 1360   CONTINUE                                                            STRM4050
C                                                                            STRM4060
      GO TO 9999                                                             STRM4070
 1370 PRINT  190, V                                                          STRM4080
      STOP                                                                   STRM4090
C                                                                            STRM4100
      GO TO 9999                                                             STRM4110
C                                                                            STRM4120
 9999 RETURN                                                                 STRM4130
C                                                                            STRM4140
   10 FORMAT  ( 1H1   /// 30X *THE FOLLOWING STREAMLINES HAVE BEEN DEV*      STRM4150
     1   *ELOPED AFTER * F7.0 * HRS. OF SITE OPERATION*  )                   STRM4160
```

```
   20  FORMAT  ( 6E15.6, 4I5 )                                                  STRM4170
   30  FORMAT  ( 1H1, 25X *STOPPED AFTER* I5 * ITERATIONS ON STREAMLINE*STRM4180
     1   I5 *TRY:* / 78X *LARGER INCREMENT(DISTME)* / 78X *GREATER INIT*STRM4190
     2   *IAL ANGLE(ANGLE)* / 78X *GENERATING MORE POINTS(MAXID).*  )     STRM4200
   40  FORMAT  ( 105X, I5 )                                                     STRM4210
   50  FORMAT  (  //// 34X *STREAMLINE NUMBER * I4 * BEGINNING AT WELL *STRM4220
     1   *(OR SPECIAL POINT) NUMBER * I4,  // 34X *INITIAL ANGLE OF THI*STRM4230
     2   *S STREAMLINE = * F6.2 * DEGREES. (IN HORIZONTAL PLANE)* // 34XSTRM4240
     3   *NUMBER OF COORDINATES PRODUCED = * I5 /  )                       STRM4250
   60  FORMAT  ( 34X *STREAMLINE HAS PASSED DESIGINATED BOUNDARIES OF A*STRM4260
     1   *NALYSIS AFTER * F12.2 * HOURS OF WELL OPERATION.* //  )          STRM4270
   70  FORMAT  ( 34X *STREAMLINE FRONT HAS NOT BROKEN THROUGH AFTER*     STRM4280
     1   F12.3 * HOURS OF WELL OPERATION.* //  )                          STRM4290
   80  FORMAT  ( 34X *STREAMLINE FRONT HAS BROKEN THROUGH AT WELL NUMBE*STRM4300
     1   *R * I4 * AFTER * F12.2 * HOURS OF WELL OPERATION.* //  )        STRM4310
   90  FORMAT  ( 2X, A8, 13(2X, F7.2) )                                         STRM4320
  100  FORMAT  ( 2X,  /  )                                                      STRM4330
  110  FORMAT  ( 34X *TWO DIMENSIONAL AREA SWEPT BY STREAMLINE NO. * I3 STRM4340
     1   * = * F10.2 * SQ.FT.*  )                                          STRM4350
  120  FORMAT  ( 3F10.2, 2E15.7, F10.2, F5.0, F10.2, 2F5.0 )                    STRM4360
  130  FORMAT  ( 3F10.5, 2E15.7 )                                               STRM4370
  140  FORMAT  (  ///// , 30X,  *BREAKDOWN OF FLUID DISPERSION FROM INP*STRM4380
     1   *UT WELL NO. * I3,  * AFTER * F8.1 * HRS. OF WELL OPERATION* //STRM4390
     2   )                                                                 STRM4400
  150  FORMAT  ( 30X *FLUID INPUT AT WELL NO. * I3 * OUTPUT AT WELL NO.*STRM4410
     1   * *  )                                                           STRM4420
  160  FORMAT  ( 74X, I3, 3X, E15.5 * GAL.*  )                           STRM4430
  170  FORMAT  (  / 30X,  *PERCENT OF WELL NO. * I3 * FLUID WITHDRAWN F*STRM4440
     1   *ROM AQUIFER * F5.1 * PERCENT* / , 30X *PERCENT OF WELL NO. *  STRM4450
     2   I3 * FLUID NOT WITHDRAWN FROM AQUIFER * F5.1 * PERCENT*  )     STRM4460
  180  FORMAT  (  // 30X *TWO DIMENSIONAL AREA SWEPT BY WELL NO. * I3   STRM4470
     1   * = * F10.2 * SQ.FT.*  )                                          STRM4480
  190  FORMAT  (  //// 20X *SIMULATION STOPPED--CSTRM ARRAY SIZE SET AT*STRM4490
     1   I8 * IS INSUFFICIENT AND MUST BE INCREASED*  )                    STRM4500
                                                                            STRM4510
C                                                                           STRM4520
       END
```

```
C////////////////////////////////////////////////////////////////////////////TMEC  10
      SUBROUTINE  TMECONF (DELPART, R, U, IRUN)                               TMEC  20
C***** THIS ROUTINE DETERMINES THE LEACHING PARAMETERS OF TIME DEP.           TMEC  30
C***** FLOW  NON-LEAKY AQUIFER SITUATION. PARTIAL OR TOTAL                   TMEC  40
C***** PENETRATION OF AQUIFER IS SPECIFIED.                                  TMEC  50
      COMMON  /PIE/  PI                                                       TMEC  60
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN                    TMEC  70
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME              TMEC  80
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,         TMEC  90
     1  N1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM                      TMEC 100
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                     TMEC 110
      DIMENSION   DELPART (50), ITME (10), R (50), SUM (50), TIME (10),       TMEC 120
     1  U (50), WELL (9,50)                                                   TMEC 130
      REAL        K, KX, KZ                                                   TMEC 140
      KEY  =  0                                                               TMEC 150
        DO 1000  I  =  1, NUMWL                                               TMEC 160
        DELPART (I)  =  0.0                                                   TMEC 170
1000    SUM (I)  =  0.0                                                       TMEC 180
        DO 1060  I  =  1, NUMWL                                               TMEC 190
        IF (WELL(5,I) .GE. TIME(IRUN) .OR. WELL(6,I) .LT. TIME(IRUN))         TMEC 200
     1    GO TO 1060                                                         TMEC 210
        C  =  WELL (7, I) / (4.0 * PI * KX * B)                               TMEC 220
        IF (WELL(2,I)-WELL(1,I) .LT. B)  GO TO 1010                           TMEC 230
        DELPART (I)  =  C * WU (U (I))                                        TMEC 240
        GO TO 1060                                                            TMEC 250
1010    IF  (WELL(2, I) .LE. 0)  WELL (2, I)  =  .0001                        TMEC 260
        SUMN  =  1.0                                                          TMEC 270
1020    ARG1  =  SUMN * PI * R (I) / B                                        TMEC 280
        ARG2  =  ARG1 *  * 2 / 4.0                                            TMEC 290
        A1  =  COS (SUMN * PI * Z / B)                                        TMEC 300
        A2  =  SIN (SUMN * PI * WELL(2, I) / B)                               TMEC 310
        A3  =  SIN (SUMN * PI * WELL(1, I) / B)                               TMEC 320
        A4  =  1. / SUMN                                                      TMEC 330
        A5  =  WU1 (I, U, ARG1, ARG2)                                         TMEC 340
        PART  =  A1 * (A2 - A3) * A4 * A5                                     TMEC 350
        SUMN  =  SUMN + 1.0                                                   TMEC 360
        SUM (I)  =  SUM (I) + PART                                            TMEC 370
        IF (ABS(PART) .LE. TERM)  GO TO 1030                                  TMEC 380
        KEY  =  0                                                             TMEC 390
        IF (SUMN .GT. ITEREND)  GO TO 1040                                    TMEC 400
        GO TO 1020                                                            TMEC 410
1030    KEY  =  KEY + 1                                                       TMEC 420
        IF (KEY .GT. 3)  GO TO 1050                                           TMEC 430
        GO TO 1020                                                            TMEC 440
1040    XP  =  X                                                              TMEC 450
        YP  =  Y                                                              TMEC 460
        CALL CNVRT (XP, YP, DMJR)                                             TMEC 470
        XP  =  XP * SIZE / 1000. + XMIN                                       TMEC 480
        YP  =  YP * SIZE / 1000. + YMIN                                       TMEC 490
        PRINT   10, SUMN, XP, YP, SUM (I), PART                               TMEC 500
1050    D1  =  WU (U(I))                                                      TMEC 510
        D2  =  B / (PI * (WELL(2, I) - WELL(1, I))) * SUM (I)                 TMEC 520
```

```
          DELPART (I)  =  C * (D1 + D2)                                    TMEC 530
 1060     CONTINUE                                                         TMEC 540
C                                                                          TMEC 550
       GO TO 9999                                                          TMEC 560
C                                                                          TMEC 570
       GO TO 9999                                                          TMEC 580
C                                                                          TMEC 590
 9999  RETURN                                                              TMEC 600
C                                                                          TMEC 610
   10  FORMAT  ( 10X *SERIES DOES NOT CONVERGE SUFFICIENTLY AFTER * F5.0TMEC 620
     1   * ITERATIONS FOR THE POINT X=* F5.2, 2X *Y=* F5.2 / 10X         TMEC 630
     2   *FINAL SUM =* E15.5 / 10X *FINAL TERM =* E15.5 )                TMEC 640
C                                                                          TMEC 650
       END                                                                 TMEC 660
```

```
C////////////////////////////////////////////////////////////////////////////TMEL  10
      SUBROUTINE  TMELKY (DELPART, R, U, IRUN)                                TMEL  20
C***** LEACHING PARAMETERS OF TIME DEPENDENT FLOW.LEAKY AQUIFER                TMEL  30
C***** SITUATION.  PARTIALLY OR TOTALLY PENETRATING OPTIONS AVAILABLE.         TMEL  40
      COMMON  /PIE/  PI                                                       TMEL  50
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN                    TMEL  60
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME              TMEL  70
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,         TMEL  80
     1   K1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM                     TMEL  90
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                     TMEL 100
      DIMENSION   DELPART (50), ITME (10), R (50), SUM (50), TIME (10),       TMEL 110
     1   U (50), WELL (9,50)                                                  TMEL 120
      REAL        K, KPRIME, KX, KZ                                           TMEL 130
      KEY  =  0                                                               TMEL 140
        DO 1000  I  =  1, NUMWL                                               TMEL 150
        DELPART (I)  =  0.0                                                   TMEL 160
1000    SUM (I)  =  0.0                                                       TMEL 170
        DO 1060  I  =  1, NUMWL                                               TMEL 180
        IF (WELL(5,I) .GE. TIME(IRUN) .OR. WELL(6,I) .LT. TIME(IRUN))        TMEL 190
     1    GO TO 1060                                                         TMEL 200
        BB  =  SQRT (KX * B * BPRIME / KPRIME)                                TMEL 210
        PRT1  =  R (I) / BB                                                   TMEL 220
        PRT2  =  PRT1 *  * 2 / 4.0                                            TMEL 230
        C  =  WELL (7, I) / (4.0 * PI * KX * B)                               TMEL 240
        IF (WELL(2,I)-WELL(1,I) .LT. B)  GO TO 1010                           TMEL 250
        DELPART (I)  =  C * WU1 (I, U, PRT1, PRT2)                            TMEL 260
        GO TO 1060                                                            TMEL 270
1010    IF  (WELL(2, I) .LE. 0)  WELL (2, I)  =  .0001                        TMEL 280
        SUMN  =  1.0                                                          TMEL 290
1020    ARG1  =  SQRT (PRT1 *  * 2 + (SUMN * PI * R(I) / B) *  * 2)           TMEL 300
        ARG2  =  ARG1 *  * 2 / 4.0                                            TMEL 310
        PART  =  COS (SUMN * PI * Z / B) * (SIN (SUMN * PI * WELL(2, I)TMEL 320
     1    / B) - SIN (SUMN * PI * WELL (1, I) / B)) * (1. / SUMN) * WU1TMEL 330
     2    (I, U, ARG1, ARG2)                                                 TMEL 340
        SUMN  =  SUMN + 1.0                                                   TMEL 350
        SUM (I)  =  SUM (I) + PART                                            TMEL 360
        IF (ABS(PART) .LE. TERM)  GO TO 1030                                  TMEL 370
        KEY  =  0                                                             TMEL 380
        IF (SUMN .GT. ITEREND)  GO TO 1040                                    TMEL 390
        GO TO 1020                                                            TMEL 400
1030    KEY  =  KEY + 1                                                       TMEL 410
        IF (KEY .GT. 3)  GO TO 1050                                           TMEL 420
        GO TO 1020                                                            TMEL 430
1040    XP  =  X                                                              TMEL 440
        YP  =  Y                                                              TMEL 450
        CALL CNVRT (XP, YP, DMJR)                                             TMEL 460
        XP  =  XP * SIZE / 1000. + XMIN                                       TMEL 470
        YP  =  YP * SIZE / 1000. + YMIN                                       TMEL 480
        PRINT   10, SUMN, XP, YP, SUM (I), PART                               TMEL 490
1050    D1  =  WU1 (I, U, PRT1, PRT2)                                         TMEL 500
        D2  =  2.0 * B / (PI * (WELL(2, I) - WELL(1, I))) * SUM (I)           TMEL 510
        DELPART (I)  =  C * (D1 + D2)                                         TMEL 520
```

```
 1060    CONTINUE                                                        TMEL 530
C                                                                        TMEL 540
       GO TO 9999                                                        TMEL 550
C                                                                        TMEL 560
       GO TO 9999                                                        TMEL 570
C                                                                        TMEL 580
 9999  RETURN                                                           TMEL 590
C                                                                        TMEL 600
   10  FORMAT  ( 10X *SERIES DOES NOT CONVERGE SUFFICIENTLY AFTER * F5.0TMEL 610
     1   * ITERATIONS FOR THE POINT X=* F5.2, 2X *Y=* F5.2 / 10X        TMEL 620
     2   *FINAL SUM =* E15.5 / 10X *FINAL TERM =* E15.5 )               TMEL 630
C                                                                        TMEL 640
       END                                                               TMEL 650
```

```
C////////////////////////////////////////////////////////////////////////SSLE  10
      SUBROUTINE  SSLEAKY (DELPART, R, IRUN)                             SSLE  20
C***** THIS ROUTINE DETERMINES THE LEACHING PARAMETERS OF TIME INDEP.     SSLE  30
C***** FLOW IN A LEAKY AQUIFER.  PARTIAL PENETRATION OF WELLS INTO        SSLE  40
C***** THE AQUIFER IS SPECIFIED BY PNDEPTH.                               SSLE  50
      COMMON  /PIE/  PI                                                  SSLE  60
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN               SSLE  70
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME         SSLE  80
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,    SSLE  90
     1  K1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM                 SSLE 100
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                SSLE 110
      DIMENSION   DELPART (50), ITME (10), R (50), SUM (50), TIME (10),  SSLE 120
     1  WELL (9,50)                                                      SSLE 130
      REAL        K, KPRIME, KX, KZ                                      SSLE 140
      KEY  =  0                                                          SSLE 150
        DO 1000  I  =  1, NUMWL                                          SSLE 160
        DELPART (I)  =  0.0                                              SSLE 170
1000    SUM (I)  =  0.0                                                  SSLE 180
        DO 1060  I  =  1, NUMWL                                          SSLE 190
        BB  =  SQRT (KX * B * BPRIME / KPRIME)                           SSLE 200
        IF (WELL(5,I) .GE. TIME(IRUN) .OR. WELL(6,I) .LT. TIME(IRUN))    SSLE 210
     1    GO TO 1060                                                    SSLE 220
        C  =  WELL (7, I) / (2.0 * PI * KX * B)                          SSLE 230
        PRT1  =  R (I) / BB                                              SSLE 240
        IF (WELL(2,I)-WELL(1,I) .LT. B)  GO TO 1010                      SSLE 250
        DELPART (I)  =  C * BESSK (PRT1)                                 SSLE 260
        GO TO 1060                                                       SSLE 270
1010    IF  (WELL(2, I) .LE. 0)  WELL (2, I)  =  .0001                   SSLE 280
        SUMN  =  1.0                                                     SSLE 290
1020    ARG  =  SQRT (PRT1 *  * 2 + (SUMN * PI * R(I) / B) *  * 2)       SSLE 300
        PART  =  COS (SUMN * PI * Z / B) * (SIN (SUMN * PI * WELL(2, I)) SSLE 310
     1    / B) - SIN (SUMN * PI * WELL (1, I) / B)) * (1. / SUMN) *      SSLE 320
     2    BESSK (ARG)                                                   SSLE 330
        SUMN  =  SUMN + 1.0                                              SSLE 340
        SUM (I)  =  SUM (I) + PART                                       SSLE 350
        IF (ABS(PART) .LE. TERM)  GO TO 1030                             SSLE 360
        KEY  =  0                                                        SSLE 370
        IF (SUMN .GT. ITEREND)  GO TO 1040                               SSLE 380
        GO TO 1020                                                       SSLE 390
1030    KEY  =  KEY + 1                                                  SSLE 400
        IF (KEY .GT. 3)  GO TO 1050                                      SSLE 410
        GO TO 1020                                                       SSLE 420
1040    XP  =  X                                                         SSLE 430
        YP  =  Y                                                         SSLE 440
        CALL CNVRT (XP, YP, DMJR)                                        SSLE 450
        XP  =  XP * SIZE / 1000. + XMIN                                  SSLE 460
        YP  =  YP * SIZE / 1000. + YMIN                                  SSLE 470
        PRINT   10, SUMN, XP, YP, SUM (I), PART                          SSLE 480
1050    DELPART (I)  =  C * (BESSK (PRT1) + 2. * (B / (PI * (WELL(2, I)) SSLE 490
     1    - WELL(1, I)))) * SUM (I))                                    SSLE 500
1060    CONTINUE                                                         SSLE 510
C                                                                        SSLE 520
```

```
      GO TO 9999                                                        SSLE 530
C                                                                       SSLE 540
      GO TO 9999                                                        SSLE 550
C                                                                       SSLE 560
 9999 RETURN                                                            SSLE 570
C                                                                       SSLE 580
   10 FORMAT  ( 10X *SERIES DOES NOT CONVERGE SUFFICIENTLY AFTER * F5.0SSLE 590
     1   * ITERATIONS FOR THE POINT X=* F5.2, 2X *Y=* F5.2 / 10X       SSLE 600
     2   *FINAL SUM =* E15.5 / 10X *FINAL TERM =* E15.5 )              SSLE 610
                                                                       SSLE 620
C                                                                       SSLE 630
      END
```

```
C////////////////////////////////////////////////////////////////////////////SSCO  10
      SUBROUTINE  SSCONF (DELPART, R, IRUN)                                  SSCO  20
C***** LEACHING SITUATION INVOLVING TIME DEPENDENT FLOW, NON-LEAKY           SSCO  30
C***** AQUIFER, PARTIAL OR TOTAL PENETRATION.                                SSCO  40
C     STEADY STATE FLOW, CONFINED AQUIFER.                                   SSCO  50
C     DEPTH OF PENETRATION = PNDEPTH                                         SSCO  60
      COMMON  /PIE/  PI                                                      SSCO  70
C                                                                            SSCO  80
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME             SSCO  90
      COMMON  /LIMITS/  XMIN, XMAX, YMIN, YMAX, SIZE, SMIN                   SSCO 100
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,        SSCO 110
     1   K1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM                    SSCO 120
      COMMON  /PERM/  KX, KY, KMJR, KMNR, DMJR, K, ZK, KZ                    SSCO 130
      DIMENSION   DELPART (50), ITME (10), R (50), SUM (50), TIME (10),      SSCO 140
     1   WELL (9,50)                                                         SSCO 150
      REAL        K, KX, KZ                                                  SSCO 160
      RE  =  300. / SIZE * 1000.                                             SSCO 170
        DO 1000  I  =  1, NUMWL                                              SSCO 180
        DELPART (I)  =  0.0                                                  SSCO 190
1000    SUM (I)  =  0.0                                                      SSCO 200
        DO 1090  I  =  1, NUMWL                                              SSCO 210
        KEY  =  0                                                            SSCO 220
        IF (WELL(5,I) .GE. TIME(IRUN) .OR. WELL(6,I) .LT. TIME(IRUN))        SSCO 230
     1    GO TO 1090                                                        SSCO 240
        C  =  WELL (7, I) / (2.0 * PI * KX * B)                              SSCO 250
        IF (WELL(2,I)-WELL(1,I) .LT. B)  GO TO 1010                          SSCO 260
        DELPART (I)  =  C * ALOG (RE / R (I))                                SSCO 270
        GO TO 1090                                                           SSCO 280
1010    IF  (WELL(2, I) .LE. 0)  WELL (2, I)  =  .0001                       SSCO 290
        SUMN  =  1.0                                                         SSCO 300
1020    ARG1  =  PI * R (I) * SUMN / B                                       SSCO 310
1030    PART  =  COS (SUMN * PI * Z / B) * (SIN (SUMN * PI * WELL(2, I)      SSCO 320
     1    / B) - SIN (SUMN * PI * WELL (1, I) / B)) * (1.0 / SUMN) *         SSCO 330
     2    BESSK (ARG1)                                                      SSCO 340
1040    CONTINUE                                                             SSCO 350
1050    SUMN  =  SUMN + 1.                                                   SSCO 360
        SUM (I)  =  SUM (I) + PART                                           SSCO 370
        IF (ABS(PART) .LE. TERM)  GO TO 1060                                 SSCO 380
        KEY  =  0                                                            SSCO 390
        IF (SUMN .GT. ITEREND)  GO TO 1070                                   SSCO 400
        GO TO 1020                                                           SSCO 410
1060    KEY  =  KEY + 1                                                      SSCO 420
        IF (KEY .GT. 3)  GO TO 1080                                          SSCO 430
        GO TO 1020                                                           SSCO 440
1070    XP  =  X                                                             SSCO 450
        YP  =  Y                                                             SSCO 460
        CALL CNVRT (XP, YP, DMJR)                                            SSCO 470
        XP  =  XP * SIZE / 1000. + XMIN                                      SSCO 480
        YP  =  YP * SIZE / 1000. + YMIN                                      SSCO 490
        PRINT   10, SUMN, XP, YP, SUM (I), PART                              SSCO 500
1080    D1  =  ALOG (RE / R(I))                                              SSCO 510
        D2  =  2. * B / (PI * (WELL(2, I) - WELL(1, I))) * SUM (I)           SSCO 520
```

```
          DELPART (I)  =  C * (D1 + D2)                                    SSC0 530
 1090     CONTINUE                                                         SSC0 540
C                                                                          SSC0 550
       GO TO 9999                                                          SSC0 560
C                                                                          SSC0 570
       GO TO 9999                                                          SSC0 580
C                                                                          SSC0 590
 9999  RETURN                                                             SSC0 600
C                                                                          SSC0 610
   10  FORMAT  ( 10X *SERIES DOES NOT CONVERGE SUFFICIENTLY AFTER * F5.0SSC0 620
     1   * ITERATIONS FOR THE POINT X = * F10.2, 2X *Y = * F10.2 / 10X  SSC0 630
     2   *FINAL SUM =* E15.5 / 10X *FINAL TERM =* E15.5 )                 SSC0 640
                                                                           SSC0 650
C                                                                          SSC0 660
       END
```

```
C////////////////////////////////////////////////////////////////////////////WU     10
      FUNCTION  WU (WUIN)                                                     WU     20
C***** CALCULATION OF WELL DRAWDOWN SOLUTION OF PARTIAL DIFFERENTIAL          WU     30
C***** EQUATION FOR TIME DEPENDENT FLOW IN A NON-LEAKY AQUIFER.              WU     40
C***** REPRESENTED AS CONVERGENT INFINITE SERIES.                            WU     50
      IF (WUIN .GT. 1.)  GO TO 1010                                           WU     60
 1000 WU  =   - .57721566 + .99999193 * WUIN - .24991055 * WUIN * WUIN       WU     70
     1   + .05519968 * WUIN *  * 3 - .00976004 * WUIN *  * 4 + .00107857WU     80
     2   * WUIN *  * 5 - ALOG (WUIN)                                         WU     90
      GO TO 9999                                                              WU    100
 1010 WU  =   (WUIN *  * 4 + 8.5733287401 * WUIN *  * 3 + 18.0590169730WU    110
     1   * WUIN * WUIN + 8.6347608925 * WUIN + .2677737343) / (WUIN *  *WU    120
     2   4 + 9.5733223454 * WUIN *  * 3 + 25.6329561486 * WUIN *  * 2 + WU    130
     3   21.0996530827 * WUIN + 3.9584969228) / (WUIN * EXP(WUIN))           WU    140
 9999 RETURN                                                                  WU    150
      END                                                                     WU    160
```

```
C////////////////////////////////////////////////////////////////////////WU1   10
      FUNCTION  WU1 (J, U, PRT1, PRT2)                                    WU1   20
C***** CALCULATION OF WELL DRAWDOWN SOLUTION OF PARTIAL DIFFERENTIAL       WU1   30
C***** EQUATION FOR TIME DEPENDENT FLOW IN A LEAKY AQUIFER.               WU1   40
      DIMENSION   U (50)                                                  WU1   50
      PRT4  =  PRT2 / U (J)                                               WU1   60
      IF (PRT4 .LT. 100)  GO TO 1000                                      WU1   70
      WU1  =  2.0 * BESSK (PRT1)                                          WU1   80
      GO TO 1040                                                          WU1   90
 1000 PRT5  =  EXP ( - PRT4)                                              WU1  100
      PRT6  =  .5772 + ALOG (U(J)) + WU (U (J)) - U (J)                   WU1  110
      PRT7  =  U (J) * ( (BESSI(PRT1) - 1.) / PRT2)                       WU1  120
      PRT10  =  0.                                                        WU1  130
        DO 1030  L  =  1, 25                                              WU1  140
          DO 1030  M  =  1, L                                             WU1  150
          IPRT8  =  L - M + 1                                             WU1  160
          IPRT9  =  L + 2                                                 WU1  170
          FACT8  =  1.                                                    WU1  180
          FACT9  =  1.                                                    WU1  190
            DO 1010  ICOUNT  =  1, IPRT8                                  WU1  200
 1010       FACT8  =  FACT8 * FLOAT (ICOUNT)                             WU1  210
            DO 1020  ICOUNT  =  1, IPRT9                                  WU1  220
 1020       FACT9  =  FACT9 * FLOAT (ICOUNT)                             WU1  230
 1030     PRT10  =  PRT10 + ((( - 1) *  * (L + M)) * FACT8) / (FACT9 *WU1  240
     1      * 2)) * (PRT2 *  * M) * (U(J) *  * (L - M))                  WU1  250
      A1  =  BESSK (PRT1)                                                 WU1  260
      A2  =  BESSI (PRT1)                                                 WU1  270
      A3  =  WU (PRT4)                                                    WU1  280
      A4  =   (PRT6 + PRT7 - (U(J) *  * 2) * PRT10)                       WU1  290
      WU1  =  2. * A1 - A2 * A3 + PRT5 * A4                               WU1  300
 1040 CONTINUE                                                            WU1  310
 9999 RETURN                                                              WU1  320
      END                                                                 WU1  330
```

```
      SUBROUTINE  CNVRT (X, Y, THETA)                                    CNVR  10
      COMMON  /PIE/  PI                                                  CNVR  20
      R  =  SQRT (X * * 2 + Y * * 2)                                     CNVR  30
      IF (R .EQ. 0.0)  GO TO 9999                                        CNVR  40
      PSI  =  ACOS (X / R)                                               CNVR  50
      IF  (Y .LE. 0.0)  PSI  =  2. * PI - PSI                            CNVR  60
      ZETA  =  PSI - THETA                                               CNVR  70
      IF  (ZETA .GE. 2. * PI)  ZETA  =  ZETA - 2.0 * PI                  CNVR  80
      IF  (ZETA .LT. 0.)  ZETA  =  ZETA + 2. * PI                        CNVR  90
      X  =  R * COS (ZETA)                                               CNVR 100
      Y  =  R * SIN (ZETA)                                               CNVR 110
9999  RETURN                                                             CNVR 120
      END                                                                CNVR 130
```

```
      SUBROUTINE  FILEDT (L1, IRUN, ITOT, CTLPT, NUM.OT)                    FILE  10
C     THIS ROUTINE EDITS CONTOUR DATA PRIOR TO OUTPUT TO TAPE12.            FILE  20
      DIMENSION   CTLPT (2,100), DAT (5), RDAT (5)                          FILE  30
      REWIND 13                                                             FILE  40
      IC  =  0                                                             FILE  50
      XSMX  =   - 9.E9                                                      FILE  60
      XSMN  =  9.E9                                                         FILE  70
      YSMX  =   - 9.E9                                                      FILE  80
      YSMN  =  9.E9                                                         FILE  90
      VMAX  =   - 9.E9                                                      FILE 100
      HMAX  =   - 9.E9                                                      FILE 110
      VMIN  =  9.E9                                                         FILE 120
      HMIN  =  9.E9                                                         FILE 130
        DO 1010  J  =  1, L1                                                FILE 140
        READ (13,   10)   (DAT(I), I=1, 5)                                  FILE 150
          DO 1000  K  =  1, NUMTOT                                          FILE 160
          R  =  SQRT ((DAT(1) - CTLPT(1, K)) *  * 2 + (DAT(2) -             FILE 170
     1      CTLPT(2, K)) *  * 2)                                           FILE 180
          IF (R .LE. 2.0)  GO TO 1010                                       FILE 190
1000      CONTINUE                                                          FILE 200
        XSMX  =  AMAX1 (XSMX, DAT(1))                                       FILE 210
        XSMN  =  AMIN1 (XSMN, DAT(1))                                       FILE 220
        YSMX  =  AMAX1 (YSMX, DAT(2))                                       FILE 230
        YSMN  =  AMIN1 (YSMN, DAT(2))                                       FILE 240
        IF (J .GT. L1-NUMTOT)  GO TO 1010                                   FILE 250
        VMAX  =  AMAX1 (VMAX, DAT(4))                                       FILE 260
        VMIN  =  AMIN1 (VMIN, DAT(4))                                       FILE 270
        HMAX  =  AMAX1 (HMAX, DAT(5))                                       FILE 280
        HMIN  =  AMIN1 (HMIN, DAT(5))                                       FILE 290
1010    CONTINUE                                                            FILE 300
      REWIND 13                                                             FILE 310
        DO 1030  J  =  1, L1                                                FILE 320
        READ (13,   10)   (DAT(I), I=1, 5)                                  FILE 330
          DO 1020  K  =  1, NUMTOT                                          FILE 340
          R  =  SQRT ((DAT(1) - CTLPT(1, K)) *  * 2 + (DAT(2) -             FILE 350
     1      CTLPT(2, K)) *  * 2)                                           FILE 360
          IF (R .LE. 2.0)  GO TO 1030                                       FILE 370
1020      CONTINUE                                                          FILE 380
        RDAT (3)  =   (DAT (4) - VMIN) / (VMAX - VMIN)                      FILE 390
        RDAT (4)  =   (DAT (5) - HMIN) / (HMAX - HMIN)                      FILE 400
        WRITE (12,   20)   (DAT(I), I=1, 2), (RDAT (I), I=3, 4)             FILE 410
        IC  =  IC + 1                                                       FILE 420
1030    CONTINUE                                                            FILE 430
      ENDST  =  999999998.                                                  FILE 440
```

```
      WRITE (12,   30)  ENDST                                                    FILE 460
      IF (IRUN .LT. ITOT)  GO TO 1040                                            FILE 470
      WRITE (12,   30)  ENDDT                                                    FILE 480
      REWIND 12                                                                  FILE 490
 1040 PRINT   40, IC, XSMX, XSMN, YSMX, YSMN, VMAX, VMIN, HMAX, HMIN             FILE 500
 9999 RETURN                                                                     FILE 510
C                                                                               FILE 520
   10 FORMAT  ( 3F10.5, 2E15.7 )                                                FILE 530
   20 FORMAT  ( 4F10.5 )                                                        FILE 540
   30 FORMAT  ( F10.0 )                                                         FILE 550
   40 FORMAT  (  // 30X, I5, 1X,  *HEAD AND VELOCITY VALUES FOR CONTOU*FILE 560
     1   *R MAP CONSTRUCTION  OUTPUT TO TAPE 12.* ///// 50X * MAXIMUM*  FILE 570
     2   8X *MINIMUM* // 38X *X-COORD* 2X, E12.5, 3X, E12.5 / 38X       FILE 580
     3   *Y-COORD* 2XE12.5, 3X, E12.5 / 37X *VELOCITY* 2XE12.5, 3X,     FILE 590
     4   E12.5 / 34X *HEAD CHANGE* 2X, E12.5, 3X, E12.5 )               FILE 600
C                                                                               FILE 610
      END                                                                       FILE 620
```

```
      SUBROUTINE  VFLOW (CSTRM, STREAM, IRUN, SIZE, KX, KZ, J, W, I, K,VFLO  10
     1   IFLAG)                                                        VFLO  20
      COMMON  /CONST/  ITME, S, B, T1, T2, GAMMA, N, ALPHA, BETA, N1,  VFLO  30
     1   K1, SS, ITOT, AQR, BPRIME, KPRIME, ITEREND, TERM              VFLO  40
      COMMON  /DATA/  X, Y, Z, WELL, NUMWL, RADIUS, NLINES, TIME       VFLO  50
C***** THIS ROUTINE CALCULATES VERTICAL, FLUID VELOCITY IN              VFLO  60
C***** PARTIALLY SCREENED AQUIFER CASES.                               VFLO  70
      DIMENSION   CSTRM (3000,5), ITME (10), STREAM (900,10), TIME     VFLO  80
     1   (10), WELL (9,50), ZLIST (2)                                  VFLO  90
      INTEGER     DISTME, W                                            VFLO 100
      REAL        KX, KZ, N                                            VFLO 110
      SCRN1  =  WELL (2, W) - WELL (1, W)                              VFLO 120
      SCRN2  =  WELL (2, K) - WELL (1, K)                              VFLO 130
      ZLOC  =   (Z - WELL(1, W)) * SCRN1                               VFLO 140
      Z2  =  ZLOC * SCRN2 + WELL (1, K)                                VFLO 150
      STREAM (1, 8)  =  RADIUS                                         VFLO 160
      RINC  =  0.0                                                     VFLO 170
      N  =  2                                                          VFLO 180
      CSTRM (1, 3)  =  Z                                               VFLO 190
      STREAM (1, 3)  =  Z / SQRT (KX / KZ) * SIZE / 1000.              VFLO 200
      KFLAG  =  0                                                      VFLO 210
      NCSPT  =  J - 1                                                  VFLO 220
      TMEFLO  =  0.0                                                   VFLO 230
        DO 1030  L  =  1, NCSPT                                        VFLO 240
        X  =  CSTRM (L, 1)                                             VFLO 250
        Y  =  CSTRM (L, 2)                                             VFLO 260
        TMEFLO  =  TMEFLO + CSTRM (L, 5)                               VFLO 270
        FOFT  =  GAMMA * (ALPHA + N * BETA) / (4.0 * TMEFLO)           VFLO 280
          DO 1000  IZ  =  1, 2                                         VFLO 290
          Z  =  CSTRM (L, 3) + 2. / 3. * IZ * 1. / SIZE - 1. / SIZE    VFLO 300
1000      CALL DELTAH (ZLIST(IZ), FOFT, IRUN)                          VFLO 310
        ZPART  =   (ZLIST(1) - ZLIST(2)) / (2. / 3. * 1. / SIZE)       VFLO 320
        VZ  =   - KZ * ZPART / N                                       VFLO 330
        ZDST  =  VZ * CSTRM (L, 5)                                     VFLO 340
        CSTRM (L + 1, 3)  =  CSTRM (L, 3) + ZDST                       VFLO 350
        GO TO 1020                                                     VFLO 360
        IF (IFLAG .NE. 3)  GO TO 1020                                  VFLO 370
        IF (KFLAG .NE. 1)  GO TO 1010                                  VFLO 380
        CSTRM (L + 1, 3)  =  Z2                                        VFLO 390
        GO TO 1020                                                     VFLO 400
1010    IF (SQRT((CSTRM(L+1,1)-WELL(3,K))**2+(CSTRM(L+1,2)-WELL(4,K))**VFLO 410
     1    2) .GE. 3.0*1000./SIZE)  GO TO 1020                         VFLO 420
        IF  (CSTRM(L + 1, 3) .LE. WELL(2, K) .AND. CSTRM(L + 1, 3) .GE.VFLO 430
     1    WELL(1, K))  KFLAG  =  1                                     VFLO 440
1020    CONTINUE                                                       VFLO 450
        RINC  =  SQRT ((CSTRM(L + 1, 1) - CSTRM(L, 1)) *  * 2 +        VFLO 460
     1    (CSTRM(L + 1, 2) - CSTRM(L, 2)) *  * 2)                      VFLO 470
        IF (CSTRM(L+1,4) .NE. 1.0)  GO TO 1030                         VFLO 480
        STREAM (M, 3)  =  CSTRM (L + 1, 3) / SQRT (KX / KZ) * SIZE /   VFLO 490
     1    1000.                                                        VFLO 500
        STREAM (M, 8)  =  STREAM (M - 1, 8) + RINC * SIZE / 1000.      VFLO 510
        RINC  =  0.0                                                   VFLO 520
```

```
        STREAM (M, 9)  =  WELL (8, W)                                    VFLO 530
        STREAM (M, 10)  =  I                                             VFLO 540
        M  =  M + 1                                                      VFLO 550
1030    CONTINUE                                                         VFLO 560
      GO TO 1050                                                         VFLO 570
      IF (IFLAG .NE. 3)  GO TO 1060                                      VFLO 580
      IF (CSTRM(NCSPT,3) .GT. WELL(1,K))  GO TO 1040                     VFLO 590
      STREAM (M, 3)  =  WELL (1, K) * SIZE / 1000.                       VFLO 600
      GO TO 1060                                                         VFLO 610
1040  IF (CSTRM(NCSPT,3) .LT. WELL(2,K))  GO TO 1050                     VFLO 620
      STREAM (M, 3)  =  WELL (2, K) * SIZE / 1000.                       VFLO 630
      GO TO 1060                                                         VFLO 640
1050  STREAM (M, 3)  =  CSTRM (NCSPT, 3) / SQRT (KX / KZ) * SIZE /       VFLO 650
     1   1000.                                                           VFLO 660
1060  CONTINUE                                                           VFLO 670
      STREAM (M, 9)  =  WELL (8, W)                                      VFLO 680
      STREAM (M, 10)  =  I                                               VFLO 690
      Z  =  CSTRM (1, 3)                                                 VFLO 700
      RETURN                                                             VFLO 710
      END                                                                VFLO 720
```

☆U.S. GOVERNMENT PRINTING OFFICE: 1980-603-102/106

INT.-BU.OF MINES,PGH.,PA. 24833

best copy available